N° d'ordre

T H E S E

présentée à

L'UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR DE 3ème CYCLE - MATHEMATIQUES APPLIQUEES

par

Françoise VEILLON

QUELQUES NOUVELLES METHODES POUR LE CALCUL NUMERIQUE DE LA TRANSFORMEE INVERSE DE LAPLACE

Thèse soutenue le 11 mars 1972 devant la Commission d'Examen :

MM. J. KUNTZMANN Président

N. GASTINEL Examinateur

P.J. LAURENT Examinateur

N° d'ordre

T H E S E

présentée à

L'UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR DE 3ème CYCLE - MATHEMATIQUES APPLIQUEES

par

Françoise VEILLON

QUELQUES NOUVELLES METHODES POUR LE CALCUL NUMERIQUE DE LA TRANSFORMEE INVERSE DE LAPLACE

Thèse soutenue le 11 mars 1972 devant la Commission d'Examen :

MM. J. KUNTZMANN	Président
N. GASTINEL	Examinateur
P.J. LAURENT	Examinateur

UNIV.
LEIDEN
BIBL.

UNIVERSITE SCIENTIFIQUE
ET MEDICALE DE GRENOBLE

LISTE DES PROFESSEURS

Président : Monsieur Michel SOUTIF
Vice-Président : Monsieur Gabriel CAU

PROFESSEURS TITULAIRES

MM.	ANGLES D'AURIAC Paul	Mécanique des fluides
	ARNAUD Georges	Clinique des maladies infectieuses
	ARNAUD Paul	Chimie
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale
	BENOIT Jean	Radioélectricité
	BESSON Jean	Electrochimie
	BEZES Henri	Chirurgie générale
	BLAMBERT Maurice	Mathématiques Pures
	BOLLIET Louis	Informatique (IUT B)
	BONNET Georges	Electrotechnique
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET-EYMARD Joseph	Pathologie médicale
	BONNIER Etienne	Electrochimie Electrométallurgie
	BOUCHERLE André	Chimie et Toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BRAVARD Yves	Géographie
	BRISSONNEAU Pierre	Physique du Solide
	BUYLE-BODIN Maurice	Electronique
	CABANAC Jean	Pathologie chirurgicale
	CABANEL Guy	Clinique rhumatologique et hydrologie
	CALAS François	Anatomie
	CARRAZ Gilbert	Biologie animale et pharmacodynamie
	CAU Gabriel	Médecine légale et Toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques Pures
	CHARACHON Robert	Oto-Rhino-Laryngologie
	CHATEAU Robert	Thérapeutique
	CHENE Marcel	Chimie papetière
	COEUR André	Pharmacie chimique
	CONTAMIN Robert	Clinique gynécologique
	COUDERC Pierre	Anatomie Pathologique
	CRAYA Antoine	Mécanique
Mme	DEBELMAS Anne-Marie	Matière médicale
MM.	DEBELMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DESSAUX Georges	Physiologie animale
	DODU Jacques	Mécanique appliquée
	DREYFUS Bernard	Thermodynamique
	DUCROS Pierre	Cristallographie
	DUGOIS Pierre	Clinique de Dermatologie et Syphiligraphie
	FAU René	Clinique neuro-psychiatrique
	FELICI Noël	Electrostatique
	GAGNAIRE Didier	Chimie physique
	GALLISSOT François	Mathématiques Pures
	GALVANI Octave	Mathématiques Pures

MM. GASTINEL Noël	Analyse numérique
GERBER Robert	Mathématiques Pures
GIRAUD Pierre	Géologie
KLEIN Joseph	Mathématiques Pures
Mme KOFLER Lucie	Botanique et Physiologie végétale
MM. KOSZUL Jean-Louis	Mathématiques Pures
KRAVTCHENKO Julien	Mécanique
KUNTZMANN Jean	Mathématiques Appliquées
LACAZE Albert	Thermodynamique
LACHARME Jean	Biologie végétale
LATREILLE René	Chirurgie générale
LATURAZE Jean	Biochimie pharmaceutique
LAURENT Pierre	Mathématiques Appliquées
LEDRU Jean	Clinique médicale B
LLIBOUTRY Louis	Géophysique
LOUP Jean	Géographie
Mle LUTZ Elisabeth	Mathématiques Pures
MALGRANGE Bernard	Mathématiques Pures
MALINAS Yves	Clinique obstétricale
MARTIN-NOEL Pierre	Seméiologie médicale
MASSEPORT Jean	Géographie
MAZARE Yves	Clinique médicale A
MICHEL Robert	Minéralogie et Pétrographie
MOURIQUAND Claude	Histologie
MOUSSA André	Chimie nucléaire
NEEL Louis	Physique du Solide
OZENDA Paul	Botanique
PAUTHENET René	Electrotechnique
PAYAN Jean-Jacques	Mathématiques Pures
PEBAY-PEYROULA Jean-Claude	Physique
PERRET René	Servomécanismes
PILLET Emile	Physique industrielle
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
REULOS René	Physique industrielle
RINALDI Renaud	Physique
ROGET Jean	Clinique de pédiatrie et de puériculture
SANTON Lucien	Mécanique
SEIGNEURIN Raymond	Microbiologie et Hygiène
SENGEL Philippe	Zoologie
SILBERT Robert	Mécanique des fluides
SOUTIF Michel	Physique générale
TANCHE Maurice	Physiologie
TRAYNARD Philippe	Chimie générale
VAILLAND François	Zoologie
VAUQUOIS Bernard	Calcul électronique
Mme VERAIN Alice	Pharmacie galénique
M. VERAIN André	Physique
Mme VEYRET Germaine	Géographie
MM. VEYRET Paul	Géographie
VIGNAIS Pierre	Biochimie médicale
YOCCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM. BULLEMER Bernhard	Physique
RADHAKRISHNA Pidatala	Thermodynamique

PROFESSEURS SANS CHAIRE

MM.	AUBERT Guy	Physique
	BEAUDOING André	Pédiatrie
	BERTRANDIAS Jean-Paul	Mathématiques Appliquées
	BIARES Jean-Pierre	Mécanique
	BONNETAIN Lucien	Chimie minérale
Mme	BONNIER Jane	Chimie générale
MM.	CARLIER Georges	Biologie végétale
	COHEN Joseph	Electrotechnique
	COUMES André	Radioélectricité
	DEPASSEL Roger	Mécanique des Fluides
	DEPORTES Charles	Chimie minérale
	DESRE Pierre	Métallurgie
	DOLIQUE Jean-Michel	Physique des Plasmas
	GAUTHIER Yves	Sciences biologiques
	GEINDRE Michel	Electroradiologie
	GIDON Paul	Géologie et Minéralogie
	GLENAT René	Chimie organique
	HACQUES Gérard	Calcul numérique
	JANIN Bernard	Géographie
Mme	KAHANE Josette	Physique
MM.	MULLER Jean-Michel	Thérapeutique
	PERRIAUX Jean-Jacques	Géologie et minéralogie
	POULOUJADOFF Michel	Electrotechnique
	REBECQ Jacques	Biologie (CUS)
	REVOL Michel	Urologie
	REYMOND Jean-Charles	Chirurgie générale
	ROBERT André	Chimie papetière
	SARRAZIN Roger	Anatomie et chirurgie
	SARROT-REYNAULD Jean	Géologie
	SIBILLE Robert	Construction Mécanique
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale
M.	VALENTIN Jacques	Physique nucléaire

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

Mle	AGNIUS-DELORD Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBLARD Pierre	Dermatologie
	AMBROISE-THOMAS Pierre	Parasitologie
	ARMAND Yves	Chimie
	BEGUIN Claude	Chimie organique
	BELORIZKY Elie	Physique
	BENZAKEN Claude	Mathématiques Appliquées
Mme	BERTRANDIAS Françoise	Mathématiques Pures
MM.	BLIMAN Samuel	Electronique (EIE)
	BLOCH Daniel	Electrotechnique
Mme	BOUCHE Liane	Mathématiques (CUS)
MM.	BOUCHET Yves	Anatomie
	BOUSSARD Jean-Claude	Mathématiques Appliquées
	BOUVARD Maurice	Mécanique des Fluides
	BRIERE Georges	Physique expérimentale
	BRODEAU François	Mathématiques (IUT B)
	BRUGEL Lucien	Energétique
	BUISSON Roger	Physique
	BUTEL Jean	Orthopédie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse

MM. CHIAVERINA Jean	Biologie appliquée (EFP)
CHIBON Pierre	Biologie animale
COHEN-ADDAD Jean-Pierre	Spectrométrie physique
COLOMB Maurice	Biochimie médicale
CONTE René	Physique
CROUZET Guy	Radiologie
DURAND Francis	Métallurgie
DUSSAUD René	Mathématiques (CUS)
Mme ETERRADOSSI Jacqueline	Physiologie
MM. FAURE Jacques	Médecine légale
GAVEND Michel	Pharmacologie
GENSAC Pierre	Botanique
GERMAIN Jean-Pierre	Mécanique
GIDON Maurice	Géologie
GRIFFITHS Michaël	Mathématiques Appliquées
GROULADE Joseph	Biochimie médicale
HOLLARD Daniel	Hématologie
HUGONOT Robert	Hygiène et Médecine préventive
IDELMAN Simon	Physiologie animale
IVANES Marcel	Electricité
JALBERT Pierre	Histologie
JOLY Jean-René	Mathématiques Pures
JOUBERT Jean-Claude	Physique du Solide
JULLIEN Pierre	Mathématiques Pures
KAHANE André	Physique générale
KUHN Gérard	Physique
Mme LAJZEROWICZ Jeannine	Physique
MM. LAJZEROWICZ Joseph	Physique
LANCIA Roland	Physique atomique
LE JUNTER Noël	Electronique
LEROY Philippe	Mathématiques
LOISEAUX Jean-Marie	Physique Nucléaire
LONGEQUEUE Jean-Pierre	Physique Nucléaire
LUU DUC Cuong	Chimie Organique
MACHE Régis	Physiologie végétale
MAGNIN Robert	Hygiène et Médecine préventive
MARECHAL Jean	Mécanique
MARTIN-BOUYER Michel	Chimie (CUS)
MAYNARD Roger	Physique du Solide
MICOUD Max	Maladies infectieuses
MOREAU René	Hydraulique (INP)
NEGRE Robert	Mécanique
PARAMELLE Bernard	Pneumologie
PECCOUD François	Analyse (IUT B)
PEFFEN René	Métallurgie
PELMONT Jean	Physiologie animale
PERRET Jean	Neurologie
PERRIN Louis	Pathologie expérimentale
PFISTER Jean-Claude	Physique du Solide
PHELIP Xavier	Rhumatologie
Mle PIERY Yvette	Biologie animale
MM. RACHAIL Michel	Médecine interne
RACINET Claude	Gynécologie et obstétrique
RICHARD Lucien	Botanique
Mme RINAUDO Marguerite	Chimie macromoléculaire
MM. ROMIER Guy	Mathématiques (IUT B)
ROUGEMONT (DE) Jacques	Neuro-Chirurgie
STIEGLITZ Paul	Anesthésiologie

MM. STOEBNER Pierre	Anatomie pathologique
VAN CUTSEM Bernard	Mathématiques Appliquées
VEILLON Gérard	Mathématiques Appliquées (INP)
VIALON Pierre	Géologie
VOOG Robert	Médecine interne
VROUSSOS Constantin	Radiologie
ZADWORNY François	Electronique

MAITRES DE CONFERENCES ASSOCIES

MM. BOUDOURIS Georges	Radioélectricité
CHEEKE John	Thermodynamique
GOLDSCHMIDT Hubert	Mathématiques
YACOUD Mahmoud	Médecine légale

CHARGES DE FONCTIONS DE MATIRES DE CONFERENCES

Mme BERIEL Hélène	Physiologie
Mme RENAUDET Jacqueline	Microbiologie

Fait le 8 MARS 1972.

Je remercie Monsieur le Professeur J. KUNTZMANN d'avoir accepté de présider le Jury de cette thèse.

J'exprime ma profonde reconnaissance à Monsieur le Professeur N. GASTINEL pour la patience dont il a fait preuve à mon égard, les conseils et encouragements qu'il m'a prodigués au cours de ce travail.

Je remercie Monsieur le Professeur P.J. LAURENT qui a bien voulu faire partie du Jury, et dont les remarques m'ont été précieuses.

Je suis très reconnaissante à Mademoiselle BICAÏS et au personnel du service de tirage qui ont apporté leur soin et leur goût à la présentation matérielle de ce travail, ainsi qu'à tous les membres du Laboratoire qui m'ont aidé dans cette tâche.

I N T R O D U C T I O N

-:-:-:-:-:-:-:-:-:-:-:-:-

CONVOLUTION DE FONCTIONS CONTINUES - PROPRIETES

OPERATEURS DE MIKUSINSKY

1 - Définition de la convolution

Soit C l'espace vectoriel des fonctions à valeurs réelles définies et continues sur l'intervalle $[0,\infty[$.

Nous noterons dorénavent par f ou $\{f(t)\}$ ces fonctions, et $f(t)$ la valeur de la fonction f au point t.

Soient a et b deux fonctions appartenant à C.

La convolution c *de* a *et de* b *est définie par :*

$$\{c(t)\} = \{\int_0^t a(t-\tau)b(\tau)d\tau\}$$

que nous noterons dorénavent :

$$c = a * b$$

Propriété

La convolution de deux fonctions appartenant à C appartient à C.

En effet, d'après les propriétés de l'intégrale, elle est définie et continue dans l'intervalle $[0,\infty[$.

II - L'espace vectoriel C est un anneau intègre par rapport aux opérations d'addition et de convolution

II.1 - L'addition est commutative et associative (C est un espace vectoriel).

II.2 - Il est trivial de démontrer que la convolution est :

commutative : $a \star b = b \star a$

associative : $(a \star b) \star c = a \star (b \star c)$

distributive par rapport à l'addition : $a \star (b+c) = (a \star b) + (a \star c)$

II.3 - Pour toute paire a et b appartenant à C il existe un élément $d \in C$ tel que :

$$a + d = b$$

Il résulte de ces trois propriétés que C est un anneau par rapport aux opérations + et $\star$.

De plus Titchmarsch [MI] a démontré que cet anneau ne contenait pas de diviseurs de zéro, c'est-à-dire que :

$$a \star b = 0$$

implique soit a = 0, soit b = 0.

Cette démonstration est longue et difficile. On la trouvera à la page 15 de la référence indiquée.

C'est cette quatrième propriété qui fait que C est un *anneau intègre* et permet de définir sur lui un *corps quotient*.

Voir [ZA] Zamansky - Introduction à l'algèbre et à l'analyse modernes p. 18

III - Construction du corps quotient et convention de notations

Considérons l'ensemble des couples [a,b] tels que a $\in$ C, b $\in$ $\{C - 0\}$, et la relation d'équivalence :

[a,b] est équivalente à [c,d] si a $\star$ d = b $\star$ c.

Il est facile de montrer que cette relation est bien reflexive, symétrique et transitive.

Notons a/b la classe d'équivalence ou ensemble des couples équivalents à [a,b] et Q l'ensemble des classes d'équivalence ainsi construites.

Nous pouvons définir sur Q une loi interne encore notée $\star$, soit :

$$a/b \star c/d = (a\star c)/(b\star d)$$

Cette loi est commutative et associative.

On peut définir pour elle un élément neutre de la forme c/c et un symétrique pour chaque élément de Q. Nous noterons b/a le symétrique de a/b, et δ l'élément neutre. Nous avons bien :

$$a/b \star b/a = (a \star b)/(b \star a) = (a \star b)/(a \star b)$$

qui est l'élément neutre. b/a est l'inverse de a/b pour la loi $\star$.

Nous pouvons maintenant établir une correspondance entre un élément de C et un élément de Q.

A a $\in$ C faisons correspondre la classe (a $\star$ c)/c. Lorsque a est fixé et que c parcourt C, on obtient des couples tous liés par la relation d'équivalence. En effet, pour [a $\star$ c,c] et [a $\star$ c',c'] on a :

$$a \star c \star c' = a \star c' \star c$$

Les classes correspondant à c et c' sont distinctes.

Enfin, à a $\star$ a' correspond la classe (a $\star$ a' $\star$ c)/c qui est bien la classe fournie par la composition des classes (a $\star$ c)/c et (a' $\star$ c')/c'.

Par conséquent, la correspondance entre a $\in$ C et (a $\star$ c)/c $\in$ Q est un *isomorphisme* entre C et la partie de Q formée par l'ensemble des classes des éléments de la forme (a $\star$ c)/c.

Pour simplifier l'écriture nous nous permettrons, à la faveur de cet isomorphisme de noter a au lieu de (a ⋆ c)/c, de même que nous noterons α et non ({α} ⋆ c)/c ({α} fonction constante de valeur α).

Pour faire de Q un corps nous prolongerons la loi d'addition sur C de la façon suivante :

$$a/b + c/d = ((a \star d) + (b \star c))/(b \star d)$$

Ceci est compatible avec l'addition dans C.
La multiplication par un scalaire λ sera prolongée de même :

$$\lambda.a/b = (\lambda.a)/b$$

Si a,b et c appartiennent à C et si $a \star b = c$ nous nous permettrons d'écrire $a = c/b$. En effet si :

$$a \star b = c$$

alors

$$(a \star d)/d \star (b \star d)/d = (c \star c)/d$$

$$(a \star d)/d \star (b \star d)/d \star d/(b \star d) = (c \star d)/d \star d/(b \star d)$$

$$(a \star d)/d = (c \star d)/d \star d/(b \star d)$$

$$(a \star d)/d = (c \star d \star d)/(d \star b \star d)$$

$$(a \star d)/d = c/b$$

Soit a = c/b avec la simplification d'écriture que nous nous sommes permise ci-dessu

IV - L'opérateur d'intégration

Soit h l'opérateur correspondant à la fonction $\{1\}$.

Nous l'appelons *opérateur d'intégration* puisque pour tout $f \in C$ nous avons :

$$h \star f = \{1\} \star f = \{\int_0^t f(\tau)d\tau\}$$

Il est facile de calculer les puissances de convolution successives de l'opérateur h :

$$h \star h = \{\frac{t}{1}\}$$

$$h \star h \star h = \{\frac{t^2}{1\times 2}\}$$

Nous admettons la notation suivante :

$$\underbrace{h \star h \ldots \star h}_{n \text{ fois}} = h^n$$

$$h^n = \{\frac{t^{n-1}}{(n-1)!}\}$$

Si nous posons 0! = 1, cette formule est vraie pour tout n supérieur ou égal à 1.

V - L'opérateur scalaire

Soit $\{\alpha\}$ la fonction constante appartenant à C de valeur α réelle. Notons la simplement α.

L'opérateur α/h est appelé *opérateur scalaire* parce que :

$$(\alpha/h) + (\beta/h) = (\alpha+\beta)/h$$

$$(\alpha/h) \star (\beta/h) = (\alpha\star\beta)/(h \star h)$$

$$(\alpha/h) \star f = (\alpha \star f)/h = \{\int_0^t \alpha\, f(\tau)d\tau\}/h$$

$$= (h \star \{\alpha f\})/h$$

$$= \alpha \star f$$

VI - L'opérateur de différenciation

Soit l'opérateur $p = \delta/h$.

On l'appelle *opérateur de différenciation* parce que, si $f \in C$ a une dérivée continue f', alors :

7.1 $$p * f = f' + f(0)$$

où $f(0) = \{f(0)\}/h$, valeur de f au point t = 0.

En effet, faisons la convolution de chaque membre de 7.1 par h. Il vient :

$$h * p * f = h * f' + \{f(0)\}$$

$$\delta * f = \{\int_0^t f'(\tau)d\tau\} + \{f(0)\}$$

$$f = \{\int_0^t f'(\tau)d\tau\} + \{f(0)\}$$

Corollaire 1

Si f a une dérivée nième continue $f^{(n)}$, alors :

$$f^{(n)} = p^n * f - p^{n-1} * f(0) - p^{n-2} * f'(0), \ldots, p * f^{(n-2)}(0) - f^{(n-1)}(0)$$

En effet, pour n = 2 on a :

$$p^2 * f = p * p * f = p * (f' + f(0))$$

$$= p * f' + p * f(0)$$

$$= f'' + f'(0) + p * f(0)$$

On démontre facilement le cas général par récurrence.

Corollaire 2

Soit α une constante. Nous avons :

$$\partial/(p-\alpha) = \{e^{\alpha t}\}$$

en effet :

$$p * \{e^{\alpha t}\} = \partial + \alpha\{e^{\alpha t}\}$$

d'où

$$\{e^{\alpha t}\} = \partial/(p-\alpha)$$

Corollaire 3

Montrons que $\partial/(p-\alpha) = \{\frac{t}{1!} e^{\alpha t}\}$

En effet :

$$\partial/(p-\alpha)^2 = \{e^{\alpha t}\} * \{e^{\alpha t}\} = \{ \int_o^t e^{\alpha(t-\tau)} e^{\alpha\tau} d\tau\}$$

$$= \{e^{\alpha t} \int_o^t d\tau\}$$

$$= \{\frac{t}{1!} e^{\alpha t}\}$$

Il est facile de montrer que d'une façon générale :

$$\partial/(p-\alpha)^n = \{\frac{t^{n-1}}{(n-1)!} e^{\alpha t}\}$$

VII - Liaison avec la transformée de Laplace

On définit ordinairement la transformée de Laplace d'une fonction f définie sur $[0,\infty[$ par :

$$L(f) = \{F(p)\} = \{\int_o^\infty e^{-pt} f(t)\, dt\}$$

dans la mesure où l'intégrale existe.

Remarquons que :

$$\{ \int_o^\infty e^{-pt} e^{\alpha t} dt\} = \frac{1}{p-\alpha}$$

et

$$\{ \int_o^\infty e^{-pt} \frac{t^{n-1}}{(n-1)!} e^{\alpha t}\, dt\} = \frac{1}{(p-\alpha)^n}$$

c'est-à-dire que pour ces deux fonctions $\{e^{\alpha t}\}$ et $\{ \frac{t^{n-1}}{(n-1)!} e^{\alpha t}\}$ la transformée de Laplace coïncide avec l'opérateur attaché à ces fonctions à condition de considérer non plus p comme un opérateur, mais comme un nombre complexe et la convolution comme une multiplication au sens usuel.

L'idée de Mikusinsky est de montrer qu'il en est ainsi pour toute fonction considérée comme opérateur et de définir la transformée de Laplace de la façon suivante :

Soit {f(t)} *considérée comme opérateur,* F(p) *son expression, si elle existe, en fonction de l'opérateur* p. *Alors* F(p) *où l'on considère* p *comme une variable complexe et la convolution comme une multiplication est par définition la transformée de Laplace de la fonction* {f(t)}.

Il faut donc montrer que :

$$F(p) = \int_0^\infty e^{-\lambda p} f(\lambda) d\lambda$$

(où p est un opérateur) a un sens et que l'expression de F(p) est identique, à l'interprétation donnée ci-dessus près, à l'expression usuelle de la transformée de Laplace de la fonction {f(t)}.

Pour ce faire, il faut poser quelques définitions et démontrer quelques théorèmes.

Remarque - Cette façon de définir la transformée de Laplace à partir d'une expression qui existe, évite d'avoir à se soucier de l'existence de l'intégrale de la définition usuelle et met l'accent sur l'aspect symbolique de cette transformée, ce qui explique en grande partie les difficultés numériques rencontrées dans la recherche par le calcul approché des transformées directes et inverse au sens de Laplace.

VIII-Fonction de classe $\mathcal{K}$

Une fonction {f(t)} *à valeurs réelles définie sur l'intervalle* $[0,\infty[$ *appartient à la classe* $\mathcal{K}$ *si :*

1°) *elle a au plus un nombre fini de points de discontinuité dans tout intervalle fini*

2°) *l'intégrale* $\int_0^t |f(\tau)| d\tau$ *a une valeur finie pour tout* t *positif.*

Si $f \in \mathcal{K}$ nous posons par définition :

$$h \star f = \{1\} \star \{f(t)\} = \{\int_0^t f(\tau)d\tau\} = a$$

La fonction a ainsi définie est continue, définie sur $[0,\infty[$ et à valeurs réelles. Nous avons donc :

$$h \star f = a$$

que nous pouvons écrire

$$f = a/h$$

Conséquence

Toute fonction de la classe $\mathcal{K}$ peut être considérée comme un opérateur puisque c'est le quotient de convolution de deux fonctions de l'espace vectoriel C.

Toutes les opérations définies sur les opérateurs sont donc définies sur les fonctions de la classe $\mathcal{K}$.

IX - Egalité de deux fonctions de classe $\mathcal{K}$ considérées comme opérateurs

Soient f et g deux fonctions appartenant à $\mathcal{K}$.
On peut écrire d'après VIII) :

$$f = a/h \quad \text{et} \quad g = b/h$$

Or a/h = b/h est équivalent à a = b, c'est-à-dire

$$\{\int_0^t |f(\tau)|d\tau\} = \{\int_0^t |g(\tau)|d\tau\} \qquad t \geq 0$$

Cette égalité est vraie quelles que soient les valeurs respectives des fonctions f et g aux points de discontinuité

Deux fonctions f et g appartenant à $\mathcal{K}$ considérées comme opérateurs sont égales si elles sont égales en tous points qui ne sont pas des points de discontinuité.

X - Puissances non entières de l'opérateur h

Nous avons vu au paragraphe IV) que

$$h^n = \{\frac{t^{n-1}}{(n-1)!}\} \qquad n \text{ entier} \tag{X.1}$$

Nous pouvons généraliser cette formule en écrivant :

$$h^\lambda = \{\frac{t^{\lambda-1}}{\Gamma(\lambda)}\}$$

pour toute valeur positive de λ. Γ est la fonction d'Euler.

Il faut considérer que ceci est une définition de l'opérateur h^λ dont (X.1) est le cas particulier où λ est entier.

Il est trivial de montrer que :

$$h^\lambda * h^\mu = h^{\lambda+\mu}$$

qui est une propriété fondamentale de la fonction puissance.

XI - Fonctions ayant une dérivée de classe $\mathcal{K}$

Une fonction $a \in C$ a une dérivée appartenant à la classe $\mathcal{K}$ si elle est dérivable dans l'intervalle $[0,\infty[$ sauf en un nombre fini de points pour tout sous intervalle fini.

THEOREME - (c'est une généralisation du théorème du §.VI et sa démonstration est identique).

Si une fonction $a \in C$ *a une dérivée* a' *appartenant à* $\mathcal{K}$, *alors :*

XI.1 $\quad p \star a = a' + a(0).$

Remarque - Ce théorème est faux si on suppose que a appartient à $\mathcal{K}$. En effet, supposons :

$$a = \begin{cases} 0 & 0 \leq t < 1 \\ 1 & 1 \leq t < \infty \end{cases}$$

Cette fonction est dérivable en tous points sauf pour t = 1 et

$$a' = \begin{cases} 0 & t \neq 1 \\ \text{non définie pour } t = 1 \end{cases}$$

D'après (IX) nous pouvons dire que a' = {0}. Comme a(0) = 0, il s'ensuivrait d'après (XI.1) que s ⋆ a = {0}, et par suite a = {0}, ce qui est faux.

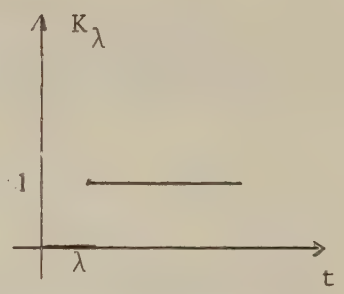
XII - Echelon et opérateur de translation

Soit

$$K_\lambda(t) = \begin{cases} 0 & 0 \leq t < \lambda \\ 1 & \lambda \leq t < \infty \end{cases}$$

et l'opérateur

$$k^\lambda = p \star K_\lambda \qquad \lambda > 0$$



Théorème

Si $\{f(t)\}$ appartient à $\mathcal{K}$ alors la valeur en t est de :

$$k^{\lambda} * f = \begin{cases} 0 & 0 \leq t < \lambda \\ f(t-\lambda) & \lambda \leq t < \infty \end{cases}$$

En effet :

$$k^{\lambda} * f = p * F_{\lambda}$$

avec

$$F_{\lambda} = \{\int_0^t K_{\lambda}(t-\tau) f(\tau) d\tau\}$$

mais

$$F_{\lambda}(t) = \begin{cases} 0 & 0 \leq t < \lambda \\ \int_0^{t-\lambda} f(\tau) d\tau & \lambda \leq t < \infty \end{cases}$$

donc

$$F'_{\lambda}(t) = \begin{cases} 0 & 0 \leq t < \lambda \\ f(t-\lambda) & \lambda \leq t < \infty \end{cases}$$

Puisque $F_{\lambda}(0) = 0$ alors :

$$k^{\lambda} * f = p * F_{\lambda} = F'_{\lambda}$$

C'est une translation du graphe de f de longueur λ dans le sens des t positifs C'est pourquoi on appelle k^{λ} *opérateur de translation.*

Ce théorème implique (c'est trivial) que :

$$k^{\lambda} * k^{\mu} * f = k^{\lambda+\mu} * f \qquad \lambda > 0,\ \mu > 0$$

Nous pouvons faire la convolution de chaque membre de cette égalité par l'opérateur δ/f. Il vient :

$$k^{\lambda} * k^{\mu} = k^{\lambda+\mu}$$

Nous avons supposé jusqu'ici λ et μ positifs. En posant $k^0 = \delta$ et $k^{-\lambda} = \delta/k^{\lambda}$ nous généralisons cela à toute valeur réelle de λ et μ.

Toutefois on peut faire la convolution de l'opérateur $k^{-\lambda}$ avec une fonction f telle que :

$$f(t) = \begin{cases} 0 & 0 \leq t < \alpha \\ 1 & \alpha \leq t < \infty \end{cases}$$

que si $0 \leq \lambda \leq \alpha$.

XIII- Fonctions à valeurs opérateurs ('operational fonctions de Mikusinski et leurs dérivées

Considérons l'opérateur h^{λ}. Puisque cet opérateur est défini quel que soit λ réel positif, nous sommes en fait en présence d'une application de $B \subset R$ (ici $B = R^{+}$) dans le corps Q des opérateurs. Appelons de telles applications *'fonctions à valeurs opérateurs'* et notons les $[f(\lambda)]$. Leur valeur au point λ est l'opérateur $\{f(\lambda)\}$.
Pour un λ positif donné, h^{λ} est une fonction de t.
En effet :

$$h^{\lambda} = \{\frac{t^{\lambda-1}}{\Gamma(\lambda)}\}$$

Il existe donc un sous-ensemble des fonctions à valeurs opérateurs, que nous noterons $[f(\lambda,t)]$ qui, a une valeur de λ font correspondre un opérateur qui est une fonction de t. Nous appelons ce sous-ensemble *'ensemble des fonctions paramétriques à valeurs opérateurs' et nous le notons P.*

Exemple : h^{λ} est paramétrique sur $[0,\infty[$.

Nous supposerons toujours que les fonctions à valeurs opérateurs prennent leurs valeurs dans $\mathcal{K}$.

Notation : La valeur au point λ de la fonction paramétrique à valeurs opérateurs $[f(\lambda,t)]$ est la fonction ou l'opérateur $\{f(\lambda,t)\}$.

XIV - Décomposition d'une fonction à valeurs opérateurs

Une fonction à valeurs opérateurs $[f(\lambda)]$ est dite décomposable dans un intervalle I fini, ouvert ou fermé, si pour tout λ appartenant à I sa valeur peut être représentée par le produit de convolution d'un opérateur q quelconque et de la valeur en λ d'une fonction appartenant à P : $[f_1(\lambda,t)]$, et que l'opérateur $\{f_1(\lambda,t)\}$ considéré comme fonction de deux variables est continu dans le domaine D défini par :

$$\begin{cases} \lambda \in I \\ t \in [0,\infty[\end{cases}$$

soit :

$$\{f(\lambda)\} = q \star \{f_1(\lambda,t)\} \quad \text{pour tout } \lambda \in I.$$

Ainsi la décomposition d'une fonction à valeurs opérateurs est ramenée à la continuité d'une fonction de deux variables.

Conséquence : Si une fonction de deux variables $\{f(\lambda,t)\}$ est continue dans le domaine D défini ci-dessus, alors la fonction paramétrique à valeurs opérateurs

$$[f(\lambda,t)]$$

est décomposable dans l'intervalle I.
En effet, nous pouvons écrire pour tout λ :

$$\{f(\lambda,t)\} = \delta \star \{f(\lambda,t)\}.$$

Il peut arriver qu'une fonction de deux variables $\{f(\lambda,t)\}$ ne soit pas continue dans le domaine D et que cependant, la fonction paramétrique à valeurs opérateurs $[f(\lambda,t)]$ soit décomposable dans l'intervalle I.
Par exemple :

$$\{K(\lambda,t)\} = \begin{cases} \{0\} & 0 \leq t < \lambda \\ \{1\} & \lambda \leq t < \infty \end{cases}$$

considérée comme fonction de deux variables est discontinue pour $\lambda = t$ et néanmoins la fonction paramétrique à valeurs opérateurs $[K(\lambda,t)]$ est décomposable puisque nou pouvons écrire pour tout $\lambda \in I$ (ici $I = R^+$) :

$$\{K(\lambda,t)\} = p \star \{k_1(\lambda,t)\}$$

avec

$$\{k_1(\lambda,t)\} = \begin{cases} \{0\} & 0 \leq t < \lambda \\ \{t-\lambda\} & \lambda \leq t < \infty \end{cases}$$

qui est continue dans le domaine D.
En effet :

$$k \star \{K(\lambda,t)\} = \{\int_0^t \{K(\lambda,\tau)\}d\tau\} = \{k_1(\lambda,t)\}$$

La fonction $[k^\lambda] = [p \star \{K(\lambda,t)\}]$ n'appartient pas à P, ses valeurs étant des opérateurs qui ne sont pas fonction de t. Cependant, elle est décomposable dans tout intervalle fini puisque nous pouvons écrire :

XIV.3 $$k^\lambda = p^2 \star \{k_1(\lambda,t)\}$$

En effet, nous avons montré que nous pouvions écrire (§.)

$$k^\lambda = k^a \star k^{\lambda-a}$$

Donc

$$k^\lambda = k^a \star p^2 \star \{k_1(\lambda-a,t)\}$$

qui est une décomposition de k^λ.

XV - Décomposition sur un intervalle infini

Une fonction à valeurs opérateurs est décomposable dans un intervalle infini I si elle est décomposable dans tout intervalle fini contenu dans I.

Exemple : k^λ est décomposable dans $]-\infty,+\infty[$

XVI - Dérivée décomposable d'une fonction à valeurs opérateurs

Une fonction à valeurs opérateurs $[f(\lambda)]$ a une dérivée décomposable $[f'(\lambda)]$ dans un intervalle fini I si l'on peut écrire pour tout $\lambda \in I$:

$$\text{XVI.1} \qquad \{f(\lambda)\} = q \ast \{f_1(\lambda,t)\} \qquad q \neq 0$$

où q est un opérateur quelconque et $\{f_1(\lambda, t)\}$ la valeur en λ d'une fonction paramétrique à valeurs opérateurs qui, considérée comme fonction de deux variables, a une dérivée partielle

$$\{\frac{\partial}{\partial\lambda} f_1(\lambda,t)\}$$

continue dans le domaine D ($\lambda \in I$, $0 \leq t < \infty$).
On écrit alors pour tout $\lambda \in I$:

$$\text{XVI.2} \qquad \{f'(\lambda)\} = q \ast \{\frac{\partial}{\partial\lambda} f_1(\lambda,t)\}$$

Montrons par exemple que $[k^\lambda]$ a une dérivée décomposable dans tout intervalle $0 \leq \lambda \leq \lambda_o$. En effet, on peut écrire pour tout $\lambda \geq 0$

$$\{k^\lambda\} = p^3 \ast h \ast \{k_1(\lambda,t)\} = p^3 \ast \{k_2(\lambda,t)\}$$

où $\{k_2(\lambda,t)\}$ est défini par :

$$\{k_2(\lambda,t)\} = \begin{cases} \{0\} & 0 \leq t < \lambda \\ \{\frac{1}{2}(t-\lambda)^2\} & 0 \leq \lambda < t \end{cases}$$

qui a une dérivée partielle $\{\frac{\partial}{\partial\lambda} k_2(\lambda,t)\}$ continue dans le domaine D ($0 \leq \lambda < \infty$, $0 \leq t < \infty$). Il s'ensuit que :

$$\text{XVI.3} \qquad \{k^\lambda\}' = p^3 \ast \{\frac{\partial}{\partial\lambda} k_2(\lambda,t)\} \qquad 0 \leq \lambda \leq \lambda_o$$

où λ_o est arbitraire positif.

Théorème

Telle qu'elle vient d'être définie, la dérivée décomposable d'une fonction à valeurs opérateurs est unique (cf. [M1] page 184).

XVII - Propriétés de la dérivée décomposable des fonctions à valeurs opérateurs

Mikusinsky a démontré les théorèmes suivants :

Théorème 1

Si la fonction à valeurs opérateurs $[f(\lambda)]$ est constante dans un intervalle I (c'est-à-dire qu'un seul et même opérateur c est associé à λ sur cet intervalle) alors sa dérivée décomposable $[f'(\lambda)] = [0]$ et réciproquement, c'est-à-dire que pour tout $\lambda \in I$

$$\{f'(\lambda)\} = \{0\}.$$

Théorème 2

Si $[f(\lambda)]$ et $[g(\lambda)]$ ont des dérivées décomposables $[f'(\lambda)]$ et $[g'(\lambda)]$ dans un intervalle I, leur somme et leur différence ont aussi des dérivées décomposables dans l'intervalle I et de plus pour tout $\lambda \in I$

$$\{f(\lambda) \pm g(\lambda)\}' = \{f'(\lambda)\} \pm \{g'(\lambda)\}.$$

Théorème 3

Avec les mêmes hypothèses qu'au théorème 2

$$\{\{f(\lambda)\}\star\{g(\lambda)\}\}' = \{f'(\lambda)\}\star\{g(\lambda)\}+\{f(\lambda)\}\star\{g'(\lambda)\}$$

Théorème 4

Si c est un opérateur arbitraire et si $[f(\lambda)]$ a une dérivée décomposable dans un intervalle I, alors $[c \star \{f(\lambda)\}]$ a une dérivée décomposable dans l'intervalle I et de plus pour tout $\lambda \in I$

$$\{c \star \{f(\lambda)\}\}' = c \star \{f'(\lambda)\}.$$

Théorème 5

Si $[g(\lambda)]$ et $[\frac{f(\lambda)}{g(\lambda)}]$ ont des dérivées décomposables dans in intervalle I alors $[f(\lambda)]$ a une dérivée décomposable dans l'intervalle I et de plus pour tout $\lambda \in I$:

$$\{\frac{f(\lambda)}{g(\lambda)}\}' = (\{f'(\lambda)\}*\{g(\lambda)\}-\{f(\lambda)\}*\{g'(\lambda)\})/\{g(\lambda)\}^2$$

Théorème 6

Si $[f(\lambda)]$ a une dérivée décomposable $[f'(\lambda)]$ sur un intervalle I et si $\{\varphi(\lambda)\}$ est une fonction numérique prenant ses valeurs sur I et ayant une dérivée continue sur un intervalle $K \subset I$, alors la fonction composée $[F(\lambda)] = [f(\varphi(\lambda))]$ a une dérivée décomposable dans l'intervalle K et de plus pour tout $\lambda \in K$:

$$\{F'(\lambda)\} = \{f'_{\varphi}(\varphi(\lambda))\}*\{\varphi'_{\lambda}(\lambda)\}.$$

XVIII-Dérivées décomposables d'ordre supérieur

La dérivée décomposable nième d'une fonction à valeurs opérateurs est la dérivée décomposable de la dérivée décomposable (n-1)ème.

XIX - Dérivée décomposable dans un intervalle infini

Une fonction $[f(\lambda)]$ a une dérivée décomposable $[f'(\lambda)]$ dans un intervalle infini I si $[f'(\lambda)]$ est dérivée décomposable de $[f(\lambda)]$ pour tout intervalle fini contenu dans I.

Nous avons montré au §.XVI que $[k^{\lambda}]$ a une dérivée décomposable dans tout intervalle fini $0 \leq \lambda \leq \lambda_o$. De plus d'après (XVI.3)

$$\{k^{\lambda}\}' = p^3 * \{-k_1(\lambda,t)\} = -p * k^{\lambda}$$

Si $[\alpha,\beta]$ est un intervalle fini arbitraire nous pouvons écrire :

$$k^{\lambda} = k^{\alpha} \star k^{\lambda-\alpha}$$

Soit

$$\ell(\lambda) = \{k^{\lambda}\}'$$

Nous pouvons écrire d'après le théorème 4 de XVII.

$$\ell(\lambda) = k^{a} \star \ell(\lambda-a)$$

Il s'ensuit l'existence d'une dérivée décomposable $[k^{\lambda}]'$ de $[k^{\lambda}]$ dans l'intervalle $[\alpha,\beta]$. Comme cet intervalle est arbitraire k^{λ} a une dérivée décomposable dans $]-\infty, +\infty[$. L'équation (XVI-3) permet d'écrire :

$$\ell(\lambda) = k^{a} \star p^{3} \star \{\frac{\partial}{\partial\lambda} k_2(\lambda-a,t)\}$$

$$= k^{a} \star p^{3} \star \{-k_1(\lambda-a,t)\}$$

$$= -p \star k^{a} \star p^{2} \star \{k_1(\lambda-a,t)\}$$

$$= -p \star k^{a} \star k^{\lambda-a}$$

$$= -p \star k^{\lambda}$$

quelle que soit la valeur de λ.

XX - Fonction exponentielle généralisée

Si ω est un nombre, alors la fonction exponentielle $x(\lambda) = e^{\lambda\omega}$ satisfait les équations

(1) $\quad x'(\lambda) = \omega x(\lambda)$

(2) $\quad x(0) = 1$

C'est la seule fonction ayant ces propriétés.

Les conditions (1) et (2) sont donc des conditions nécessaires et suffisantes pour que :

$$x(\lambda) = e^{\lambda\omega} .$$

Elles peuvent donc servir de définition de l'exponentielle. Si ω n'est plus un nombre mais un opérateur quelconque, ces conditions définissent une exponentielle généralisée que nous noterons $e^{\lambda\omega}$ par extension. La fonction a valeurs opérateurs k^{λ} satisfait les conditions

$$\{k^{\lambda}\}' = -p \star k^{\lambda} \quad \text{et} \quad k^{0} = \ell$$

Par conséquent nous pouvons écrire :

$$\{k^{\lambda}\} = \{e^{-\lambda p}\}$$

XXI -Théorème

Si une fonction $\{f(\lambda)\}$est une fonction à valeurs réelles définie sur le segment $[\lambda_1,\lambda_2]$, alors

$$\{\int_{\lambda_1}^{\lambda_2} \{e^{-\lambda p} f(\lambda)\}d\lambda\} = \begin{cases} \{f(t)\} & \lambda_1 < t < \lambda_2 \\ \{0\} & \text{ailleurs} \end{cases}$$

Pour démontrer ce théorème posons tout d'abord $\lambda_1 = 0$
Puisque $\{e^{-\lambda p}\} = p\star \{k(\lambda,t)\}$ avec

$$\{k(\lambda,t)\} = \begin{cases} \{0\} & 0 \leq t < \lambda \\ \{1\} & \lambda \leq t < \infty \end{cases}$$

nous avons :

$$\{\int_{o}^{\lambda_2} \{e^{-\lambda p} f(\lambda)\}d\lambda\} = p \star \{\int_{o}^{\lambda_2} \{k(\lambda,t) f(\lambda)\}d\lambda\}$$

$$= p \star \{\int_{o}^{t} \{g(\lambda_2,\lambda)\}d\lambda\}$$

$$= \{g(\lambda_2,t)\}$$

avec

$$\{g(\lambda,t)\} = \begin{cases} \{f(t)\} & 0 \leq t < \lambda \\ \{0\} & \lambda \leq t < \infty \end{cases}$$

Le cas où λ_1 est positif se ramène au cas précédent de la façon suivante :

$$\{\int_{\lambda_1}^{\lambda_2}\{e^{-\lambda p}f(\lambda)\}d\lambda\} = \{\int_{o}^{\lambda_2}\{e^{-\lambda p}f(\lambda)\}d\lambda\} - \{\int_{o}^{\lambda_1}\{e^{-\lambda p}f(\lambda)\}d\lambda\}$$

Comme

$$\{\int_{o}^{\infty}\{e^{-\lambda p}f(\lambda)\}d\lambda\} = \{\lim_{\beta\to\infty}\int_{o}^{\beta}\{e^{-\lambda p}f(\lambda)\}d\lambda\}$$

il s'ensuit que :

$$\{\int_{o}^{\infty}\{e^{-\lambda p}f(\lambda)\}d\lambda\} = \{f(t)\}$$

La limite existe évidemment quelle que soit la fonction $f \in \mathcal{K}$.

XXII - Suites de fonctions - Convergence univorme et presque uniforme

Une suite de fonctions $\{f_n(t)\}$ à valeurs réelles définies et bornées sur un intervalle I *converge uniformément* vers $\{f(t)\}$ dans I si il existe une suite de nombres ε_n tendant vers 0 telle que

$$|f_n(t) = f(t)| \leq \varepsilon_n$$

pour tout t appartenant à I.

Cette suite de fonctions *converge presque uniformément* vers $\{f(t)\}$ dans I (on dit aussi est localement uniformément convergente) si elle converge uniformément vers $\{f(t)\}$ dans tout sous-intervalle fini de I.

Mikusinsky [MI page 141 et suivantes] a démontré que :

Théorème 1

Si $\{f_n(t)\}$ appartient à C et si $\{f_n(t)\}$ converge presque uniformément vers $\{f(t)\}$ dans $[0,\infty[$, alors $\{f(t)\}$ appartient à C.

Théorème 2

Si f_n et g_n convergent uniformément dans $[0,t_o]$ vers f et g respectivement, alors

$$f_n * g_n$$

converge uniformément vers $f * g$ dans $[0,t_o]$

Théorème 3

Si g_n converge uniformément vers g dans $[0,t_o]$ et si f(t) est une fonction continue dans cet intervalle, alors

$$f * g_n$$

converge uniformément vers $f * g$ dans cet intervalle.

Théorème 4

Si g_n converge uniformément vers g dans $[0,t_o]$ alors $\{\int_o^t g_n(\tau)d\tau\}$ converge uniformément vers $\{\int_o^t g(\tau)d\tau\}$ dans cet intervalle.

XXIII Définition de la convergence de suites d'opérateurs ou convergence*

Une suite d'opérateurs a_n *est convergente* * *s'il existe un opérateur* $q \neq 0$ *tel que la suite* a_n/q *soit une suite de fonctions appartenant à C presque uniformément convergente sur* $[0,\infty[$. [*MI page 144*].

Exemple :

$$a_n = p^2n^2/(p^2+n^2).$$

Cette suite est convergente * parce que a_n/p^4 est une fonction appartenant à C presque uniformément convergente sur $[0,\infty[$.
En effet :

$$a_n/p^4 = n^2/(p^2 * (p^2+n^2)) = 1/p^2-1/(p^2+n^2)$$

$$= \{t- \frac{1}{n} \sin nt\}.$$

Cette fonction converge presque uniformément vers $\{t\}$ sur $[0,\infty[$.
On écrit :

$$\lim_{n\to\infty}{}^* n^2p^2/(p^2+n^2) = p^4 * \{t\} = p^2$$

soit plus généralement

$$\lim_{n\to\infty}{}^* a_n = q * \lim_{n\to\infty} a_n/q.$$

XXIV - Continuité de la transformation de Laplace

Telle qu'elle a été définie par Mikusinsky comme une application de C dans l'espace des opérateurs Q, la transformée de Laplace L est une application continue de C dans Q, *si l'on considère dans C la convergence presque uniforme sur* $[0,\infty[$ *et dans Q la convergence* *.
En effet, Mikusinsky [page 360] a montré que l'image par L d'une suite convergente de C est une suite convergente * de Q.

XXV - Continuité de la transformation inverse

Si l'on considère dans C la convergence presque uniforme sur $[0,\infty[$ et dans Q la convergence *, alors la transformation inverse de Laplace L^{-1} qui à un opérateur $\{f\}$ fait correspondre la fonction f, n'est pas continue : *en effet, il existe des suites de Q convergentes* * *dont l'image par* L^{-1} *sont des suites qui ne convergent pas presque uniformément.*

Exemples

Soit $a_n = \{\cos nt\} = p/(p^2+n^2)$

Prenons $q = p$

$$a_n/q = 1/(p^2+n^2) = \frac{1}{n} \times n/(p^2+n^2) = \{\frac{\sin nt}{n}\}$$

qui converge presque uniformément vers $\{0\}$. En effet :

$$|\frac{\sin nt}{n} - 0| \leq \frac{1}{n}$$

On montrerait de la même façon (en prenant $q = p^2$) que :

$$\lim_{n\to\infty}^{*} \{n \sin nt\} = 1$$

et (en prenant $q = p^3$) que :

$$\lim_{n\to\infty}^{*} \{n^2 \cos nt\} = p.$$

Dans ces trois exemples la suite d'opérateurs est convergente * alors que de toute évidence les suites de fonctions $\{\cos nt\}$, $\{n \sin nt\}$ et $\{n^2 \cos nt\}$ ne sont pas presque uniformément convergentes.

XXVI - Difficultés du calcul numérique de la transformation inverse

Nous dirons que deux transformées F(p) et G(p) sont infiniment voisines si sur l'intersection U de leurs domaines respectifs de convergence :

$$\operatorname{Sup}_U |F(p)-G(p)|$$

peut être rendu plus petit que tout $\varepsilon > 0$ fixé à l'avance. Toutes les difficultés rencontrées dans le calcul numérique de la transformée inverse au sens de Laplace viennent du fait que deux transformées infiniment voisines peuvent avoir des inverses aussi différentes que possible.

Par exemple considérons les fonctions $\{\sin t\}$ et $\{\sin (1+\varepsilon)t\}$ dont les transformées sont respectivement $1/(p^2+1)$ et $(1+\varepsilon)/(p+(1+\varepsilon)^2)$. Leur abscisse de convergence est 0 (maximum des parties réelles des pôles).

U est donc le $\frac{1}{2}$ plan à droite de l'axe imaginaire.

$$\sup_U \left|\frac{1}{p^2+1} - \frac{1+\varepsilon}{p^2+(1+\varepsilon)^2}\right| = \sup_U \left|\frac{p^2+(1+\varepsilon)^2-(1+\varepsilon)(p^2+1)}{(p^2+1)(p^2+(1+\varepsilon)^2)}\right|$$

$$= \sup_U \left|\frac{\varepsilon(1-p^2)+\varepsilon^2}{(p^2+1)(p^2+(1+\varepsilon)^2)}\right.$$

Cette expression tend vers 0 en même temps que ε. Or, quel que soit ε, on peut toujours trouver un sous intervalle fini I de $[0,\infty[$ tel que :

$$\sup_I |\sin t - \sin (1+\varepsilon)t| = 2.$$

CHAPITRE - 1

METHODE DE PAPOULIS UTILISANT LES POLYNOMES DE LEGENDRE

I-1 Rappel de la méthode de Papoulis pour l'inversion de la transformée de Laplace [PA] [NU] [BU].

Cette méthode est basée sur le développement de la fonction originale f(t) sur une famille de fonctions orthogonales. Ce sont les fonctions de Legendre qui sont ici utilisées.

Soit F(p) la transformée de Laplace de f(t)

$$F(p) = \int_o^\infty e^{-pt} f(t)\, dt$$

Faisons sur cette équation le changement de variable

$$t = \varphi(x) \text{ avec } x = \varphi^{-1}(t)$$

Il vient :

$$F(p) = \int_{\varphi^{-1}(0)}^{\varphi^{-1}(\infty)} e^{-p\varphi(x)} f(\varphi(x))\varphi'(x)\, dx$$

Il suffit de déterminer φ de sorte que $e^{-p\varphi(x)} \varphi'(x)$ soit une fonction simple $g_p(x)$ pour obtenir le produit scalaire

$$F(p) = \int_{\varphi^{-1}(0)}^{\varphi^{-1}(\infty)} g_p(x)\, f(\varphi(x))\, dx$$

S'il est possible de déterminer ce changement afin d'obtenir les composantes de $f(\varphi(x))$ sur un système complet de fonctions, ces composantes déterminent $f(\varphi(x))$ donc $f(t)$.

Posons $\quad -\frac{1}{\sigma} \operatorname{Ln}(x) = \varphi(x) \quad$ où $\quad x = e^{-\sigma t} \qquad (\sigma > 0)$

$$F(p) = -\int_1^0 e^{p/\sigma \operatorname{Ln}(x)} f(-\frac{1}{\sigma} \operatorname{Ln}(x)) \frac{dx}{\sigma x}$$

$$F(p) = \int_0^1 x^{(p/\sigma - 1)} f(-\frac{1}{\sigma} \operatorname{Ln}(x)) \frac{dx}{\sigma}$$

Si on veut que le système complet soit celui des puissances paires de x sur l'intervalle $[0,1]$ on ne prend que des valeurs de p telles que $\frac{p}{\sigma} - 1 = 2k$ (k entier).

On obtient alors l'équation :

$$\sigma F[(2k+1)\sigma] = \int_0^1 x^{2k} f(-\frac{1}{\sigma} \operatorname{Ln}(x)) \, dx$$

On peut poser a priori :

$$f(-\frac{1}{\sigma} \operatorname{Ln}(x)) = \sum_{\ell=0}^{\infty} C_\ell P_{2\ell}(x)$$

où les C_ℓ sont des coefficients inconnus et $P_{2\ell}$ les polynômes de Legendre de degré 2ℓ.
Il vient :

$$\sigma F[(2k+1)\sigma] = \int_0^1 x^{2k} \sum_{\ell=0}^{\infty} C_\ell P_{2\ell}(x) \, dx$$

Athanasios Papoulis a cherché à obtenir les coefficients C_i en explicitant les valeurs $\sigma R[(2k+1)\sigma]$ en fonction des C_ℓ. Il a obtenu la matrice triangulaire suivante :

$$\sigma F(\sigma) = C_o$$

$$\sigma F(3\sigma) = \frac{C_o}{3} + \frac{2C_1}{3\times 5}$$

$$(1) \qquad \sigma F[(2k+1)\sigma] = \frac{C_o}{2k+1} + \frac{2kC_1}{(2k+1)(2k+3)} \cdots \frac{2k(2k-2)\ldots 2C_k}{(2k+1)(2k+3)\ldots(4k+1)}$$

C'est cette matrice qu'il faut inverser pour obtenir l'expression des C_i en fonction des $F[(2k+1)\sigma]$.

On peut déterminer une expression simple de cette inversée par la méthode suivante, due à Gastinel : les polynômes de Legendre sur $[-1,+1]$ et de degré pair sont orthogonaux sur $[0,1]$ à un coefficient près.

En effet :

$$\int_0^1 P_{2i}(x)\, P_{2j}(x)\, dx = \frac{\partial_{ij}}{4i+1}$$

∂_{ij} = symbole de Kronecker.

Désignons par $a_{2i,2k}$ les coefficients des polynômes de Legendre de degré pair où 2i est l'ordre du polynôme et 2k le rang du coefficient.

Reprenons l'équation (3), multiplions de part et d'autre par $a_{2i,2k}$ et faisons la somme par rapport à k variant de 0 à i. Il vient :

$$(2) \qquad \sum_{k=0}^{i} a_{2i,2k}\sigma F[(2k+1)\sigma] = \int_0^1 \sum_{k=0}^{i} a_{2i,2k}\, x^{2k} \sum_{\ell=0}^{\infty} C_\ell P_{2\ell}(x)\, dx$$

Si $\ell \neq i$ l'intégrale est nulle.

Si $\ell = i$ on obtient

$$(3) \qquad \sum_{k=0}^{i} a_{2i,2k}\sigma F[(2k+1)\sigma] = \frac{C_i}{4i+1}$$

Posons $(4i+1)a_{2i,2k} = y_{ik}$. On obtient alors

$$(4) \qquad f(t) = \sum_{\ell=0}^{\infty} P_{2\ell} e^{-\sigma t} \sum_{k=0}^{\ell} y_{\ell k}\sigma F[(2k+1)\sigma]$$

On sait d'autre part que :

$$P_{2i}(x) = \sum_{k'=0}^{i} (-1)^{k'} \frac{(4i-2k')!\, x^{2(i-k')}}{2^{2i} k'!(2i-k')!(2i-2k')!}$$

$$a_{2i,2k} = \sum_{k=0}^{i} (-1)^{i-k} \frac{(2i-2k)!}{2^{2i}(i-k)!(i+k)!(2k)!}$$

Il vient alors :

$$(5) \qquad y_{ik} = (-1)^{i-k} \frac{(4i+1)(2i-2k)!}{2^{2i}(i-k)!(i+k)!(2k)!}$$

Les formules (4) et (5) résolvent complètement le problème.

I-2 - Modalités d'emploi de la méthode

L'expérience a montré que pour des fonctions croissantes et monotones, il fallait choisir $\sigma = \frac{k}{t}$.

Si on reprend la formule (4) on voit qu'il faut limiter la sommation relative à ℓ. Pratiquement on s'arrête à 13 (au delà les coefficients $y_{\ell k}$ deviennent trop grands et il faudrait prendre trop de chiffres significatifs).

La valeur de k est choisie de telle façon que e^{-k} soit la racine la plus proche de $\frac{1}{e}$ du polynôme de Legendre de degré 24. On augmente ainsi la précision de la méthode en rendant négligeable le dernier terme de la sommation.

On peut aussi limiter la sommation sur ℓ par test de précision à une valeur z telle que

$$\frac{\sum_{\ell=0}^{z} y_{\ell k}\ \sigma F[(2k+1)\sigma]}{y_{\ell z}\ \sigma F[(2z+1)\sigma]} < 10^{-6}$$

On est sur de cette façon d'avoir toujours au moins deux chiffres significatifs. On peut montrer qu'il faut que les $F[(2k+1)\sigma]$ décroissant rapidement pour que le test soit positif pour une valeur de ℓ inférieure ou égale à 13.

Il peut être intéressant de limiter le test de précision à une valeur supérieure à 10^{-6} si la fonction F(p) n'est pas facilement inversible.

CHAPITRE - 2

METHODE DE PAPOULIS UTILISANT LES POLYNOMES DE LAGUERRE

II-1 - Méthode de Papoulis utilisant les polynômes de Laguerre [PA]

Les polynômes de Laguerre sont définis par :

$$L_k(t) = e^t \frac{d^k}{dt^k} \left[\frac{t^k}{k!} e^{-t}\right].$$

En posant $\varphi_k(t) = e^{-t} L_k(t)$ on obtient facilement la transformée de $\varphi_k(t)$

$$\varphi_k(p) = \frac{p^k}{(p+1)^{k+1}}$$

Comme les dérivées de $\varphi_k(p)$ d'ordre inférieur à k sont nulles à l'origine, on doit avoir :

$$\int_0^\infty t^n \varphi_k(t)\, dt = 0 \quad \text{pour } n \leq k-1$$

Posons

$$f(t) = \sum_{k=0}^{\infty} C_k \varphi_k(t) \qquad (0)$$

il vient

$$F(p) = \sum_{k=0}^{\infty} C_k \frac{p^k}{(p+1)^{k+1}} \qquad (1)$$

En dérivant n fois le développement en série de puissance à l'origine de $\frac{1}{p+1}$ que :

$$\frac{p^k}{(p+1)^{k+1}} = p^k \sum_{n=0}^{\infty} C_{n+k}^k (-1)^n p^n \qquad (2)$$

En développant F(p) à l'origine on obtient :

$$F(p) = \sum_{k=0}^{\infty} a_k \, p^k \qquad (3)$$

Par identification des puissances égales de p dans les équations (1), (2) et (3) on obtient le système :

$$a_o = C_o$$

$$a_1 = C_1 - C_o$$

$$a_k = C_k - C_k^1 C_{k-1} + \ldots + (-1)^k C_o$$

Ce système se résoud immédiatement :

$$C_k = \sum_{i=0}^{k} C_k^j \, a_{k-j} \qquad (4)$$

Connaissant les coefficients a_k du développement en série de R(p) donné en (3), on peut déterminer grâce à (4) les coefficients intervenant en (0).

Supposons que f(t) soit approchée par la somme finie

$$f_N(t) = \sum_{k=0}^{N} C_k \, \varphi_k(t)$$

des (N+1) premiers termes de (0).
Alors les transformées $F_N(p)$ et F(p) de $f_N(t)$ et f(t) respectivement ont des dérivées égales à l'origine jusqu'à l'ordre N et

$$\int_o^{\infty} t^n \, f_N(t) \, dt = \int_o^{\infty} t^n \, f(t) \, dt \qquad n \leq N$$

C'est-à-dire que les fonctions f(t) et $f_N(t)$ ont des moments égaux jusqu'à l'ordre N.

Application - Prenons comme fonction φ_k la fonction sinus :

$$\varphi_k(\theta) = \sin (2k+1)\theta$$

alors

$$f_N(\theta) = \sum_{k=0}^{N} C_k \sin(2k+1)\theta$$

avec

$$\frac{\pi}{4} C_n = \sigma 2^{2n} F[(2n+1)\sigma] - \sum_{j=0}^{N-1} [C_{2n}^{j-1} - C_{2n}^{n-j-1}] C_j$$

II-2 - Exemples utilisant les polynômes de Laguerre

Les exemples choisis pour illustrer cette méthode ont été choisis tels que f(t) est connu de façon à pouvoir comparer f(t) à $f_n(t)$. Le choix de σ n'a pas d'importance. Pour les calculs σ a été pris égal à 0,2.

Exemple

$$F(p) = \frac{\pi}{4} \frac{1}{(p+0,2)^2+1}$$

a pour inverse

$$f(t) = \frac{\pi}{4} e^{-0,2t} \sin t$$

θ degrés	$t = -\frac{1}{\sigma} \mathrm{Ln}(\cos\theta)$	$f_N(t) \times 10^4$	$f(t) \times 10^4$	méthode de Gaver Stehfest	méthode de Dubner Abate améliorée
5	$1.906\ 10^{-2}$	398	149.138	149.1	149.138
10	$7.654\ 10^{-2}$	432	591.465	591.5	591.465
15	$1.733\ 10^{-1}$	1158	1308.45	1309	1308.45
20	$3.110\ 10^{-1}$	2511	2258.55	2259	2258.55
25	$4.919\ 10^{-1}$	3362	3361.78	3359	3361.78
30	$7.192\ 10^{-1}$	4215	4480.90	4472	4480.90
35	$9.974\ 10^{-1}$	5571	5404.73	5413	5404.73
40	1.333	6029	5846.59	5935	5846.58
45	1.733	5181	5480.82	5583	5480.81
50	2.210	4048	4052.61	3853	4052.61
55	2.779	1944	1596.52	1042	1596.52
60	3.466	-1502	-1250.75	-1385	-1250.75
65	4.306	-3272	-3049.46	-2074	-3049.47
70	5.364	-1590	-2135.15	-1225	-1235.16
75	6.758	570	929.56	- 151,9	929.56
80	8.754	694	848.17	246.5	848.21
85	$1.220\ 10^{-1}$	- 33	- 245.02	104.8	- 256.7

CHAPITRE - 3

METHODE DE REGULARISATION [RI]

Une autre façon d'aborder le problème du calcul de la transformée inverse d'une fonction au sens de Laplace est de discrétiser pour de nombreuses valeurs de p l'équation

$$F(p) = \int_o^\infty e^{-pt} f(t)\, dt$$

afin de se ramener à la résolution d'un système linéaire surdimentionné à résoudre au sens des moindres carrés.

III-1 - Discrétisation de l'opérateur d'intégration

Pour pouvoir utiliser cette méthode il faut que la fonction à intégrer soit nulle ou négligeable au delà d'une certaine valeur τ de t pour pouvoir ramener l'intégrale infinie à une intégrale finie :

$$F(p) = \int_o^\tau e^{-pt} f(t)\, dt$$

Il faut donc que :

$$\int_\tau^\infty e^{-pt} f(t)\, dt \quad \text{soit négligeable ou nulle.}$$

Le facteur d'amortissement e^{-pt} permet en général cette supposition. Alors, on découpe l'intervalle $[0,\tau]$ en q intervalles égaux et dans chacun de ces intervalles on applique par exemple une formule d'intégration de Newton Cotes d'ordre 2r+1.

L'intervalle $[0,\tau]$ comprend donc $2rq+1$ points t_j équidistants. L'abscisse de t_j est donc $\frac{j\tau}{2q}$. Confondons le point et son abscisse. Dans le ième intervalle la fonction à intégrer est remplacée par le polynôme d'interpolation

$$P_{2r+1,i}(t) = \sum_{j=2(i-1)r}^{2ir} L_j^i(t)\, e^{-pt_j}\, f(t_j)$$

avec

$$L_j^i(t) = \frac{\prod_{\substack{k=2(i-1)r \\ k\neq j}}^{2ir} (t-t_k)}{\prod_{\substack{k=2(i-1)r \\ k\neq j}}^{2ir} (t_j-t_k)} \qquad \begin{array}{l} i = 1,2,\ldots,q \\ j = 2(i-1)r \ldots 2ir \end{array}$$

Les essais qui ont été faits par Ribière utilisaient la formule de Simpson qui est un cas particulier des formules de Newton Cotes ($r = 1$).
On prend alors pour $F(p)$:

$$F(p) = \sum_{i=1}^{q} \int_{t_{2(i-1)}}^{t_{2i}} P_{3,i}(t)\, dt$$

$$= \sum_{i=1}^{q} \int_{t_{2(i-1)}}^{t_{2i}} \sum_{j=2(i-1)}^{2i} L_j^i(t)\, e^{-pt_j}\, f(t_j)\, dt$$

On montre par un calcul simple que $F(p)$ peut se mettre alors sous la forme :

$$F(p) = \sum_{j=0}^{2q} f(t_j) e^{-pt_j} A_j$$

avec :

$$A_o = \frac{\tau}{6q}$$

$$A_{2k} = 2 \times \frac{\tau}{6q} \qquad 2k < 2q$$

$$A_{2k+1} = 4 \times \frac{\tau}{6q} \qquad 2k+1 < 2q$$

$$A_{2q} = \frac{\tau}{6q}$$

III-2 - Ecriture du système linéaire

On peut écrire cette discrétisation de F(p) pour m valeurs réelles de p, m étant nettement plus grand que 2q.

On obtient alors un système surdimensionné à résoudre au sens des moindres carrés. En effet :

$$F(p_1) = \sum_{j=0}^{2q} f(t_j)\, e^{-p_1 t_j}\, A_j$$

$$\text{-------}$$

$$F(p_m) = \sum_{j=0}^{2q} f(t_j)\, e^{-p_m t_j}\, A_j$$

s'écrit

$$\frac{}{6q}\begin{vmatrix} 1 & 4e^{-p_1 t_1} & 2e^{-p_1 t_2} & \dots & e^{-p_1 t_{2q}} \\ 1 & 4e^{-p_2 t_1} & 2e^{-p_2 t_2} & \dots & e^{-p_2 t_{2q}} \\ & & & & \\ & & & & \\ & & & & \\ 1 & 4e^{-p_m t_1} & 2e^{-p_m t_2} & \dots & e^{-p_m t_{2q}} \end{vmatrix} \times \begin{vmatrix} f(t_o) \\ --- \\ \\ \\ \\ f(t_{2q}) \end{vmatrix} = \begin{vmatrix} F(p_1) \\ --- \\ \\ \\ \\ F(p_m) \end{vmatrix}$$

Symbolisons ce système par

$$Af = F$$

III-3 - Résolution du système linéaire

Résoudre le système :

$$Af = F$$

au sens des moindres carrés où A à m lignes et (2q+1) colonnes avec $m >> 2q+1$ revient à chercher un vecteur f tel que :

(1) $\quad \|F-Af\| = \|r\|$ norme euclidienne

soit minimum.

Si on suppose que les n colonnes de A sont linéairement indépendantes, (c'est notre cas), on sait que ce problème admet une solution unique vérifiant le système linéaire carré

$$A^T A f = A^T F$$

Le résidu r est en général non nul. Il est nul si et seulement si F appartient au sous-espace $\mathcal{E}(A)$ engendré par les colonnes de A (ce n'est pas souvent le cas). Mais les erreurs d'arrondi, de données et surtout le mauvais conditionnement de $A^T A$ font que le résultat est un vecteur f^* différent de f.
Ribière d'après Tichonoff [TI] a étudié la manière d'améliorer ces calculs.

Au lieu de minimiser (1) il propose d'ajouter à A un certain nombre de contraintes exprimées par une matrice S à k lignes et 2q+1 colonnes et de chercher $f \in R^{2q+1}$ minimisant

$$\left\| \begin{vmatrix} f \\ \\ 0 \end{vmatrix} - \begin{vmatrix} A \\ \\ \omega S \end{vmatrix} \tilde{f} \right\|$$

La norme est prise dans R^{m+k} - ω est à déterminer. $\tilde{f}$ est déterminé par la résolution de :

$$(A^T \mid \omega S^T) \begin{pmatrix} A \\ \\ \omega S \end{pmatrix} \tilde{f} = (A^T \mid \omega S^T) \begin{pmatrix} F \\ \\ 0 \end{pmatrix}$$

Soit

$$(A^T A + \gamma H)\tilde{f} = A^T F$$

avec

$$\gamma = \omega^2 \quad \text{et} \quad H = S^T S$$

Ceci montre que H est une matrice (2q+1, 2q+1) symétrique semi-définie positive. Ribière a montré que pour certaines valeurs de γ et de H, on peut obtenir numériquement une solution $\tilde{f}^*$ telle que :

$$\|\tilde{r}^*\| = \|A\tilde{f}^* - F\| < \|r^*\| = \|Af^* - F\|$$

Le problème des moindres carrés peut être résolu d'une autre façon.

Soit F_o la projection orthogonale de F sur (A). Le problème revient à chercher f tel que

$$Af = F_o$$

Le résidu s'écrit alors

$$r = F_o - F$$

Mais la résolution numérique du problème des moindres carrés fournit une solution $f^* \neq f$, le résidu étant alors :

$$r^* = Af^* - F = Af - F + A(f^* - f)$$

$$r^* = (F_o - F) + A(f^* - f)$$

Par définition de F_o, les vecteurs $F_o - F$ et $A(f^* - f)$ sont orthogonaux. D'où :

$$\|f^*\|^2 = \|F_o - F\|^2 + \|A(f^* - f)\|^2 \qquad (2)$$

La méthode que l'on propose vise à réduire la norme du résidu. On voit qu'on ne peut espérer que réduire le deuxième terme du second membre de (2).
Or c'est le seul terme qui reste lorsqu'on résoud numériquement :

$$Af = F$$

si $F \in \mathcal{C}(A)$
Les conclusions obtenues par Ribière et que nous donnons ci-dessous sont basées sur l'hypothèse que $F \in \mathcal{C}(A)$ et que rang(A) = 2q+1.

III-4 - Conclusions

Les essais faits par Ribière ont montré que :

a) La valeur optimum de γ est toujours voisine de ε, précision relative de la machine utilisée.

b) La méthode de décomposition QR et la méthode du gradient conjugué donnent des résultats légèrement plus précis que la méthode de Gauss.

c) Que la matrice S qui donne les meilleurs résultats est la matrice des différences secondes :

$$S = \begin{vmatrix} -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ & & & -1 & 2 & -1 & 0 \\ 0 & & & & -1 & 2 & -1 \end{vmatrix}$$

III-5 - Raffinement de la méthode

Ribière propose enfin une méthode itérative permettant de raffiner la méthode précédente.
Il s'agit de résoudre :

$$(A^T A+\gamma H) f_{q+1} = A^T F+\gamma H f_q$$

Cette méthode à l'avantage que partant de $f_o = 0$ on obtiendra dès la première itération, avec un γ convenable, une solution f_1 assez proche de la solution exacte. Les itérations suivantes permettent de réduire le résidu.

Ribière a montré que le procédé itératif convergeait et que la valeur de optimum était encore voisine de la précision relative de la machine.
Enfin, il a montré que quelle que soit la matrice A (c'est-à-dire même si rang(A) $< 2q+1$) le procédé itératif converge vers la solution minimisant

$$||Af-F||$$

si celle-ci est unique, sinon et à condition que $A^T A+H$ soit non singulière, vers la solution qui minimise à la fois

$$||Af-F|| \quad \text{et} \quad f^T H f$$

III-6 - Comparaison entre | - la méthode de régularisation
| - la méthode de Gaver Stehfest
| - la méthode de Dubner Abate améliorée

Cette comparaison a été faite sur la fonction :

$$f(t) = \begin{cases} 1 + \sin(\pi t) & 0 \leq t \leq 2 \\ 0 & t > 2 \end{cases}$$

dont la transformée est :

$$F(p) = \left(\frac{1}{p} + \frac{\pi}{p^2+\pi^2}\right)(1-e^{-2p})$$

Soit $f_R(t)$ la fonction calculée par Ribière par la méthode de Gauss avec minimisation des différences secondes. Les résultats donnés ci-dessous correspondent à $\gamma = 10^{-6}$ (qui n'est pas égal à la précision de la machine utilisée, sur 704 où $\gamma = 10^{-8}$). En effet Ribière a calculé la norme de l'erreur et sait que c'est pour $\gamma = 10^{-6}$ dans ce cas particulier que la norme de l'erreur est la plus petite.

Nous avons tout de même choisi cet exemple comme référence car il constitue les meilleurs résultats obtenus parmi toutes les variantes essayées.
$f_{GS}(t)$ est la fonction calculée avec la méthode de Gaver Stehfest en double précision.

$f_{DA}(t)$ est la fonction calculée avec la méthode de Dubner Abate améliorée en double précision.

La dernière colonne du tableau ci-après représente l'erreur relative calculée à partir des valeurs en double longueur de f(t) et de $f_{DA}(t)$.

Ces valeurs de f, f_R, f_{GS} et f_{DA} ne sont données qu'avec 4 chiffres significatifs parce que nous n'en possédions pas plus pour f_R.

Il est clair que c'est la méthode de Dubner Abate améliorée qui donne les meilleurs résultats.

t	f(t) vraie	$f_R(t)$	$f_{GS}(t)$	$f_{DA}(t)$	erreur relative sur $f_{DA}(t)$
0	1	1,009	-	-	-
0,2	1,588	1,6	1,586	1,588	$1,2\ 10^{-7}$
0,4	1,951	1,899	1,965	1,951	$1,1\ 10^{-5}$
0,6	1,951	1,918	1,974	1,949	$9,8\ 10^{-4}$
0,8	1,588	1,678	1,469	1,588	$1,4\ 10^{-4}$
1,0	1	1,061	0,904	1	$9,9\ 10^{-8}$
1,2	0,412	0,333	0,537	0,412	$1,2\ 10^{-5}$
1,4	0,049	-0,04	0,336	0,057	$1,6\ 10^{-1}$
1,6	0,049	0,115	0,226	0,042	$1,3\ 10^{-1}$
1,8	0,412	0,488	0,158	0,528	$2,8\ 10^{-1}$
2,0	1	0,831	0,111	0,476	$5,2\ 10^{-1}$

CHAPITRE - IV

METHODE DE BELLMAN [BKL]

IV-1 - Exposé de la méthode

La méthode ou plutôt les méthodes utilisées par Bellman sont basées, comme la méthode proposée par Ribière, sur une discrétisation préalable de l'équation

$$F(p) = \int_0^\infty e^{-pt} f(t)\, dt$$

Le problème initial se transforme donc en celui de la résolution d'un système linéaire :

$$Af = F$$

Bellman propose, plutôt que de résoudre brutalement ce système, qui est mal conditionné, au sens des moindres carrés en minimisant :

$$(Af-F,\ Af-F)$$

de minimiser :

$$(0) \qquad (Af-F,\ Af-F) + \lambda\varphi(f) \qquad \lambda \geq 0$$

où $\varphi(f)$ et λ sont adroitement choisis.

Remarque - La méthode de Ribière correspond à un $\varphi(f)$ particulier.

IV-2 - Cas où une approximation f_a de f est déjà connue

L'obtention d'un vecteur f_a sera étudié plus loin. Bellman propose en premier lieu de poser :

$$\varphi(f) = (f_a - f,\ f_a - f)$$

le but recherché est alors de minimiser :

$$(1) \qquad (Af - F,\ Af - F) + \lambda\,(f_a - f,\ f_a - f)$$

Bien entendu, le choix de λ est important et nous en reparlerons.
Un calcul simple montre que le vecteur qui minimise (1) s'écrit :

$$f(\lambda) = (A^T A + \lambda I)^{-1}\ (A^T F + \lambda f_a)$$

L'expérience et l'analyse montrent que $A^T A + \lambda I$ est moins mal conditionnée que A même pour de petites valeurs de λ (ce n'est pas vrai pour $\lambda = 0$).
Si λ est petit on peut espérer que $f(\lambda)$ soit une bonne approximation de la solution de

$$Af = F.$$

Il s'ensuit que l'on peut utiliser une méthode classique pour résoudre :

$$(A^T A + \lambda I) f = A^T F + \lambda f_a$$

Bellman propose cependant une autre méthode qui est un exemple de résolution numérique d'équations par procédé à plusieurs étapes de décision et qui mène à la programmation dynamique.

IV-3 - Première approche de la programmation dynamique

Pour appliquer la programmation dynamique à la minimisation de (1) nous choisissons f par un procédé à étapes de la façon suivante : nous choisissons d'abord sa $n^{ème}$ composante f_n, puis f_{n-1} etc...

Pour ce faire considérons d'abord le problème plus général de minimisation de :

$$(2) \qquad R_M(f) = [\lambda(f_1-f_{a1})^2+\lambda(f_2-f_{a2})^2+ \ldots + \lambda(f_M-f_{aM})^2+ \sum_{i=1}^{N} \Big(\sum_{j=1}^{M} a_{ij}f_j-F_i \Big)^2]$$

où $1 \leq M \leq N$ et les f_{ai} sont les composantes d'une première approximation f_a.

Si M = N nous retombons sur le problème initial.

Si M = 1 nous avons le problème simple de minimisation de

$$R_1(f) = \lambda(f_1-f_{a1})^2+ \sum_{i=1}^{N} (a_{i1}f_1-F_i)^2$$

Notre but est d'obtenir une relation de récurrence entre

$$R_M \text{ et } R_{M-1}$$

Si nous extrayons f_M, nous sommes ramenés à la minimisation d'une fonction du même type sur $f_1,\ldots,f_{M-1}$ où F est devenu $F-f_M a^{(M)}$.

Donc si nous introduisons :

$$h_M(F) = \min_{f_1 \ldots f_M} R_M(f)$$

définie pour M = 1,...,N et pour tout F nous obtenons pour $M \geq 2$ la relation de récurrence :

$$(3) \qquad h_M(F) = \min_{f_M} [\lambda(f_M-f_{aM})^2-h_{M-1}(F-f_M a^{(M)})]$$

où

$$h_1(F) = \min_{f_1} R_1(f).$$

C'est une application du principe d'optimalité.
Si la dimension de F était petite nous pourrions utiliser (3) pour calculer l'ensemble des valeurs de $h_M(F)$. Sinon il faut se servir du fait que $h_M(F)$ est une fonction quadratique de F de la forme :

$$h_M(F) = (F, Q_M F) + 2(p_M, F) + r_M$$

où Q_M est une matrice indépendante de F, p_M un vecteur indépendant de F, r_M un scalaire indépendant de F.

Des calculs laborieux permettent de calculer Q_M, p_M et r_M en fonction de Q_{M-1}, p_{M-1} et r_{M-1}.
Soit

$$Q_M = Q_{M-1} - \frac{A^{(M)}}{\lambda + K_M}$$

$$p_M = p_{M-1} - (\lambda f_{aM} + \rho_M)\alpha^{(M)}$$

$$r_M = r_{M-1} + \lambda f_{aM}^2 - \frac{(\lambda f_{aM} + \rho_M)^2}{\lambda + K_M}$$

où

$$\alpha^{(M)} = Q_{M-1} a^{(M)}$$

$$\rho_M = (p_{M-1}, a^{(M)})$$

$$K_M = (\alpha^{(M)}, a^{(M)})$$

et

$$A^{(M)} = \alpha^{(M)} \oplus \alpha^{(M)}$$

où $\oplus$ désigne le produit de Kronecker :

$$a \oplus b = (a_i b_j) \qquad i = 1, 2, \ldots, N$$

En prenant :

$$\begin{cases} Q_o = I \\ p_o = 0 \\ r_o = 0 \end{cases}$$

nous pouvons calculer Q_M, p_M et r_M pour $M \geq 1$.

IV-4 - Détermination des valeurs minimisantes

Pour obtenir le vecteur f minimisant, nous partons de Q_o, p_o et r_o pour calculer Q_M, p_M et r_M et ceci pour M = 1,2,...,N.

Puis nous déterminons f_M, f_{M-1},...,f_1

$$f_M = \frac{\lambda f_{aM} + (p_{M-1}, a^{(M)}) + (a^{(M)}, Q_{M-1}F)}{\lambda + (a^{(M)}, Q_{M-1}a^{(M)})}$$

Mais Q_{M-1} est symétrique et nous pouvons écrire :

$$(a^{(M)}, Q_{M-1}F) = (Q_{M-1}a^{(M)}, F)$$

Il vient :

$$f_M = \frac{\lambda f_{aM} + \rho_M + (\alpha^{(M)}, F)}{\lambda + K_M}$$

Remarque - $\alpha^{(M)}$, ρ_M et K_M ont déjà été calculés.

Pour calculer f_{M-1} il suffit de remplacer F par $F - f_M a^{(M-1)}$ dans cette expression, etc...

C'est ce procédé qu'utilise le programme écrit par Bellman.

Remarque - Il peut venir à l'idée de poser $\lambda = 0$ et d'utiliser cette méthode pour résoudre Af = F. Mais si A est mal conditionné, les résultats ne sont pas meilleurs que ceux donnés par les méthodes classiques.

IV-5 - Approximations successives

Soit f_a une approximation initiale. Le vecteur $f_{a,i}$ minimisant :

$$(Af-F,\ Af-F) + \lambda(f-f_a,\ f-f_a)$$

peut être considéré comme une seconde approximation...etc. Soit $f_{a,n}$ le vecteur minimisant :

$$(Af-F,\ Af-F) + \lambda(f-f_{a,n-1},\ f-f_{a,n-1}) .$$

Pour montrer que la suite des $f_{a,i}$ est convergente, explicitons

$$f_{a,n} = (A^TA+\lambda I)^{-1}A^TF + \lambda(A^TA+\lambda I)^{-1}f_{a,n-1}$$

Si A est régulière, alors A^TA est symétrique définie positive. Les valeurs propres de $A^TA+\lambda I$ sont plus grandes que λ, et les valeurs propres de $\lambda(A^TA+\lambda I)^{-1}$ sont positives et inférieures à 1. La suite des $f_{a,i}$ converge donc géométriquement.

Soit f la limite de cette suite. Alors

$$(A^TA+\lambda I)f = A^TF+\lambda f$$

Or

$$A^TAf = A^TF$$

Comme A est régulière :

$$Af = F$$

Remarquons que pour le choix de λ , nous sommes dans un dilemne.
$\lambda = 0$ donne la convergence la plus rapide. Mais si A est mal conditionnée, les résultats ne seront pas précis.
Au contraire si λ est plus grand, la convergence sera lente mais $A^TA+\lambda I$ bien conditionnée.

IV-6 - Obtention d'une approximation initiale

Si nous résolvons l'équation

$$\int_0^\infty f(t)\ e^{-pt}\ dt = F(p)$$

par quadrature, nous sommes ramenés à la résolution, comme nous l'avons vu d'un système linéaire :

$$(4) \qquad \sum_{i=1}^{N} w_i\ f(t_i)\ e^{-pt_i} = F(p) \qquad p = 1,2,\ldots,N.$$

Nous pouvons alors détourner la difficulté de la façon suivante :

(a) résolution du système (4) pour N = 5 par exemple

(b) recherche d'une courbe très lisse passant par ces 5 points ou les approchant très bien.

(c) lecture sur cette courbe de 10 points (par exemple) qui vont servir d'approximation initiale dans la méthode décrite précédemment.

IV-7 - Développement de l'approximation initiale en série de puissance

Nous savons que le vecteur $f(\lambda)$ qui minimise :

$$(Af-F,\ Af-F) + \lambda(f-f_a,\ f-f_a)$$

est donné par :

$$f(\lambda) = (A^TA+\lambda I)^{-1}(F+\lambda f_a)$$

Au voisinage de 0, $f(\lambda)$ a un développement en série de puissances :

$$f(\lambda) = f_{min} + \lambda g_1 + \lambda^2 g_2$$

De plus $f(\lambda)$ est continu et dérivable indéfiniment pour $\lambda > 0$.
Il s'ensuit que nous devons pouvoir obtenir la valeur de f_{min}, c'est-à-dire $f(0)$ à partir des valeurs de $f(\lambda)$ pour λ quelconque.

Considérons le cas où $f(\lambda)$ peut être obtenu pour $\lambda > 0$ et suffisamment grand, auquel cas $A^TA+\lambda I$ n'est pas mal conditionnée.

La valeur de f ainsi obtenue peut encore servir d'approximation initiale pour la méthode décrite précédemment.

IV-8 - Approximations successives avec lissage

La méthode des approximations successives se trouve améliorée si après un certain nombre d'itérations on lisse la solution obtenue et que l'on prenne cette solution lissée comme nouvelle approximation initiale.
Bellman ne donne pas de précision sur le lissage qu'il utilise mais indique seulement qu'il n'est pas du tout sophistiqué.
Il donne un exemple qui montre que ce lissage intermédiaire améliore beaucoup les résultats.

IV-9 - Autre méthode proposée par Bellman

Si l'on sait que la solution cherchée est raisonnablement lisse et régulière, on peut espérer alors que :

$$D_N(f) = (f_1-f_2)^2+(f_2-f_3)^2+\ldots+(f_{N-1}-f_N)^2$$

est assez petit. Les f_i sont les composantes de f.
Bellman propose donc de faire $\varphi(f) = D_N(f)$ dans (0) et d'utiliser la programmation dynamique pour minimiser :

$$(Af-F,\ Af-F)+\lambda D_N(f)$$

Pour ce faire, considérons le problème de minimisation de :

$$\sum_{i=1}^{M}\Big(\sum_{j=2}^{N} a_{ij}f_j-z_i\Big)^2+\lambda\left[(f_N-f_{N-1})^2+\ldots+(f_2-c)^2\right]$$

sur $f_2,\ldots,f_N$ où les z_i et c sont des paramètres.

Appelons $h_2(z,c)$ ce minimum pour N fixé.

C'est une forme quadratique en z et c.

Introduisons la séquence :

$$h_k(z,c) = \min_{f_r} \Big[\sum_{i=1}^{M} \Big(\sum_{j=k}^{N} a_{ij} f_j - z_i \Big)^2 + \lambda \big[(f_N - f_{N-1})^2 + \ldots + (f_k - c)^2 \big] \Big]$$

où

$$r = k, k+1, \ldots, N$$

$$k = 2, 3, \ldots, N$$

Il vient :

$$h_N(z,c) = \min_{f_N} \Big[\sum_{i=1}^{M} (a_{iN} f_N - z_i)^2 + \lambda (f_N - c)^2 \Big] \tag{5}$$

et le principe d'optimalité donne la relation de récurrence :

$$h_k(z,c) = \min_{f_k} \big[\lambda (f_k - c)^2 + h_{k+1}(z - a^k f_k, f_k) \big]$$

pour $k = 2,3,\ldots N-1$ où

$$a^k = \begin{vmatrix} a_{1k} \\ a_{2k} \\ \vdots \\ a_{Mk} \end{vmatrix}$$

En se servant du fait que chaque fonction $h_k(z,c)$ est de la forme

$$h_k(z,c) = (z, Q_k z) + 2(z, P_k)c + r_k c^2 \tag{6}$$

et de la relation de récurrence nous voulons trouver des relations de récurrence sur les Q, les P et les r.

Pour résoudre pour f_N nous égalons à zéro la dérivée partielle par rapport à f_N du terme entre crochets de (5).

Il vient

$$0 = \lambda(f_N - c) + (a^N, a^N f_N - z)$$

$$f_N = \frac{\lambda c + (a^N, z)}{\lambda + (a^N, a^N)}$$

ce qui donne la valeur de $h_N(z,c)$

$$h_N(z,c) = -\lambda c(f_N - c) - (z, a^N f_N - z)$$

$$= \lambda c^2 + (z, Iz) - \lambda c \left[\frac{\lambda c + (a^N, z)}{\lambda + (a^N, a^N)}\right] - \left(z, a^N \frac{\lambda c + (a^N, z)}{\lambda + (a^N, a^N)}\right)$$

$$= c^2\left(\lambda - \frac{\lambda^2}{\lambda + (a^N, a^N)}\right) + 2c\left(\frac{-\lambda(a^N, z)}{\lambda + (a^N, a^N)}\right) + \left(z, \left[I - \frac{A_N}{\lambda + (a^N, a^N)}\right] z\right)$$

où $A_N = a^N \oplus a^N$.

Donc

$$Q_N = I - \frac{A_N}{\lambda + (a^N, a^N)}$$

$$P_N = -\frac{\lambda a^N}{\lambda + (a^N, a^N)}$$

$$r_N = \frac{\lambda(a^N, a^N)}{\lambda + (a^N, a^N)}$$

Pour obtenir les relations de récurrence, calculons f_{N-1} et $h_{N-1}(z,c)$ de la même façon :

$$f_{N-1}(z,c) = \operatorname*{Min}_{f_{N-1}} \left[\lambda(f_{N-1} - c)^2 + h_N(z - a^{N-1} f_{N-1}, f_{N-1})\right]$$

$$= \operatorname*{Min}_{f_{N-1}} \left[\lambda(f_{N-1} - c)^2 + (z - a^{N-1} f_{N-1}, Q_N(z - a^{N-1} f_{N-1}))\right.$$

$$\left. + 2(P_N, (z - a^{N-1} f_{N-1}))f_{N-1} + r_N f_{N-1}^2\right] .$$

Annulons la dérivée partielle par rapport à f_{N-1} de $h_{N-1}(z,c)$

$$0 = \lambda(f_{N-1}-c)-(a^{N-1},Q_N(z-a^{N-1}f_{N-1}))$$

$$+(P_N,z-a^{N-1}f_{N-1}))-(P_N,a^{N-1}f_{N-1})+r_N f_{N-1}$$

$$= f_{N-1}[\lambda+(a^{N-1},Q_N a^{N-1})-2(P_N,a^{N-1})+r_N]$$

$$-[\lambda c+(a^{N-1},Q_N z)-(P_N,z)]$$

$$f_{N-1} = \frac{\lambda c+(a^{N-1},Q_N z)-(P_N,z)}{\lambda+(a^{N-1},Q_N a^{N-1})-2(P_N,a^{N-1})+r_N}$$

Posons :

$$\alpha^N = Q_N a^{N-1}$$

$$\rho^N = (P_N,a^{N-1})$$

$$K_N = (a^{N-1},\alpha^N)$$

Il vient :

$$f_{N-1} = \frac{\lambda c+(\alpha^N,z)-(P_N,z)}{\lambda+K_N-2\rho_N+r_N}$$

$$h_{N-1}(z,c) = (z,Q_{N-1}z)+2(P_N,z)c+r_{N-1}c^2 = \lambda c^2-\lambda c f_N+(z,Q_N z)-(z,Q_N a^{N-1}f_{N-1}$$

$$+(P_N f_N,z)$$

$$= (z,Q_N z)+\lambda c^2-\frac{\lambda^2 c^2+\lambda c(\alpha^N,z)-\lambda c(P_N,z)}{\lambda+K_N-2\rho_N+r_N}-(z,\alpha^N\frac{\lambda c+(\alpha^N,z)-(P_N,z}{\lambda+K_N-2\rho_N+r_N}$$

$$+(P_N,z\frac{\lambda c+(\alpha^N,z)-(P_N,z)}{\lambda+K_N-2\rho_N+r_N})$$

$$h_{N-1}(z,c) = (z,Q_{N-1}z)+2(P_{N-1},z)c+r_{N-1}c^2$$

$$= (z,Q_Nz)- \frac{1}{\lambda+K_N-2\rho_N+r_N} (\alpha^N-P_N,z)^2$$

$$+ 2c \frac{\lambda}{\lambda+K_N-2\rho_N+r_N} (P_N-a^N,z)+c^2\lambda(\frac{K_N-2\rho_N+r_N}{\lambda+K_N-2\rho_N+r_N})$$

Il vient donc :

$$Q_{N-1} = Q_N - \frac{1}{\lambda+K_N-2\rho_N+r_N} (\alpha^N-P_N) \oplus (\alpha^N-P_N)$$

$$P_{N-1} = \frac{\lambda}{\lambda+K_N-2\rho_N+r_N} (P_N-\alpha^N)$$

$$r_{N-1} = \frac{\lambda(K_N-2\rho_N+r_N)}{\lambda+K_N-2\rho_N+r_N}$$

Après calcul des P, Q et r, on peut calculer f_1 en minimisant la forme quadratique (6) sur c.

En remplaçant c par f_1 et z par $F-a^1f_1$, on calcule f_2 etc...

IV-10 - Tableau comparatif des résultats obtenus

sur la fonction $f(t) = e^{-t/2}$ par les méthodes

- de Bellman ($f_B(t)$)
- de Dubner Abate améliorée ($f_{DA}(t)$)
- de Gaver Stehfest ($f_{GS}(t)$)

t	f(t) vraie	$f_B(t)$	$f_{DA}(t)$	$f_{GS}(t)$
4.140186	0.126174	0.120527	0.126174	0.126489
2.501126	0.286329	0.288195	0.286329	0.286264
1.643438	0.439675	0.439084	0.439675	0.439603
1.085084	0.581269	0.581308	0.581269	0.581249
0.693147	0.707107	0.707318	0.707107	0.707105
0.412298	0.813712	0.813401	0.813712	0.813709
0.214821	0.898157	0.898482	0.898158	0.898154
0.085541	0.958131	0.957847	0.958135	0.958128
0.016048	0.992008	0.992205	0.992015	0.992008

C'est la méthode de Dubner Abate améliorée qui donne les meilleurs résultats.

Cependant la méthode de Gaver Stehfest devient très précise pour t petit.

CHAPITRE - V

METHODE DE GAVER STEHFEST [ST] [GA]

V-1 - Exposé théorique

Cette méthode a été développée par Gaver et programmée par Stehfest.

Soit une fonction f(t) et sa transformée de Laplace F(p).

On peut exprimer d'une façon approchée f(t) à partir de l'espérance de f(t) par rapport à la loi de probabilité suivante :

$$\begin{cases} f_n(a,t) = a \dfrac{2n!}{n!(n-1)!} (1-e^{-at})^n e^{-nat} \\ a > 0 \end{cases}$$

L'espérance mathématique de f(t) par rapport à cette loi est :

$$E_n(f) = \int_o^\infty f(t)\ f_n(a,t)\ dt$$

$$E_n(f) = \int_o^\infty f(t)\ a \frac{2n!}{n!(n-1)!} (1-e^{-at})^n\ e^{-nat}\ dt$$

$$E_n(f) = a \frac{2n!}{n!(n-1)!} \int_o^\infty f(t) \sum_{i=0}^{n} C_n^i (-1)^i e^{-at} i e^{-at} n\ dt$$

$$E_n(f) = a \frac{2n!}{n!(n-1)!} \sum_{i=0}^{n} C_n^i (-1)^i \int_o^\infty f(t)\ e^{-at(n+i)}\ dt$$

$$(1) \qquad E_n(f) = a \frac{2n!}{n!\ (n-1)!} \sum_{i=0}^{n} C_n^i (-1)^i\ F((n+i)a)$$

La loi de probabilité à les propriétés suivantes :

a) $$\int_0^\infty f_n(a,t)\, dt = 1$$

b) Le mode de $f_n(a,t) = \frac{Ln(2)}{a}$ (c'est le zéro de la dérivée de f_n).

c) La variance de f(t) par rapport à cette loi de probabilité est :

$$var_t = \int_0^\infty [f(t)-E(f(t))]^2 f_n(a,t)dt = \frac{1}{a^2} \sum_{i=0}^{n} \frac{1}{(n+i)^2}$$

Ces trois propriétés impliquent que $E_n(f)$ converge vers $f(\frac{Ln(2)}{a})$ quand n tend vers l'infini.

$E_n(f)$ a pour développement asymptotique :

(1) $$E_n(f) \simeq f(\frac{Ln(2)}{a}) + \frac{\alpha_1}{n} + \frac{\alpha_2}{n^2} + \frac{\alpha_3}{n^3} + \ldots$$

En se reportant à l'équation (1) on voit que N valeurs de F permettent de calculer $E_{n-1}(f)$ qui est une approximation de $f(\frac{Ln(2)}{a})$.

Mais on peut obtenir une meilleure approximation de cette valeur par combinaison linéaire de $E_1(f)$, $E_2(f)$,...$E_{N/2}(f)$.

Posons :

(2) $$\sum_{i=1}^{K} x_i(K) \frac{1}{(\frac{N}{2}+1-i)^k} = \delta_{ko}$$

pour $k = 0,1,\ldots,K-1$ et $K = \frac{N}{2}$,

où δ_{ko} est le symbole de Kronecker (égal à 1 pour k = 0 et nul pour les autres valeurs).

(1) est un système linéaire dont le déterminant est un déterminant de Vandermonde. Il vient :

(3) $$x_i(K) = \frac{(-1)^{i-1}}{K!} C_K^i\, i\, (\frac{N}{2}+1-i)^{K-1}$$

Montrons que dans ces conditions :

$$S = \sum_{i=1}^{K} x_i(K)\, E_{\frac{N}{2}+1-i}(f) = f\left(\frac{Ln(2)}{a}\right) + (-1)^{K+1}\,\alpha\,\frac{(\frac{N}{2}-K)!}{\frac{N}{2}!} + o\left(\frac{(\frac{N}{2}-K)!}{(\frac{N}{2})!}\right)$$

$$S = \sum_{i=1}^{K} \frac{(-1)^{i-1}}{K!}\, C_K^i\, i\,(\tfrac{N}{2}+1-i)^{K-1}\left[f\left(\frac{Ln(2)}{a}\right) + \frac{\alpha_1}{\frac{N}{2}+1-i} + \frac{\alpha_2}{(\frac{N}{2}+1-i)^2} \ldots\right]$$

en vertu de (1) et (3).

En se servant de (2) il vient :

$$S = f\left(\frac{Ln(2)}{a}\right) + \sum_{i=1}^{K} \frac{(-1)^{i-1}}{K!}\, C_K^i\, i\left[\frac{\alpha_K}{\frac{N}{2}+1-i} + \frac{\alpha_{K+1}}{(\frac{N}{2}+1-i)^2} + \ldots\right]$$

Posons

$$Z_k = \sum_{i=1}^{k} \frac{(-1)^{i-1}}{i!}\, C_k^i\, i\, \frac{\alpha_i}{\frac{N}{2}+1-i}$$

$$Z_k = \sum_{j=0}^{k-1} \frac{(-1)^j}{(k-1)!}\, C_{k-1}^j\, \frac{\alpha_k}{\frac{N}{2}-j}$$

Supposons que :

$$Z_{k+1} = \sum_{j=0}^{k} \frac{(-1)^j}{k!}\, C_k^j\, \frac{1}{\frac{N}{2}-j} = (-1)^k\, \frac{(\frac{N}{2}-k-1)!}{(\frac{N}{2})!}$$

C'est vrai pour k = 0 et k = 1.

Montrons que c'est encore vrai pour Z_{k+2}

$$Z_{k+2} = \sum_{j=0}^{k+1} \frac{(-1)^j}{(k+1)!}\, C_{k+1}^j\, \frac{1}{\frac{N}{2}-j}$$

$$= \sum_{j=0}^{k} \frac{(-1)^j}{k!}\, C_k^j\, \frac{1}{(\frac{N}{2}-j)(k+1-j)} + \frac{(-1)^{k+1}}{(k+1)!}\, \frac{1}{\frac{N}{2}-k-1}$$

Or

$$\frac{1}{(\frac{N}{2}-j)(k+1-j)} = \frac{1}{\frac{N}{2}-k-1}\left[\frac{1}{\frac{N}{2}-j} + \frac{1}{k+1-j}\right]$$

$$Z_{k+2} = \frac{1}{\frac{N}{2}-k-1}\left[-\sum_{j=0}^{k}\frac{(-1)^j}{k!}C_k^j\,\frac{1}{\frac{N}{2}-j} + \sum_{j=0}^{k}\frac{(-1)^j}{k!}\,\frac{C_k^j}{k+1-j} + \frac{(-1)^{k+1}}{(k+1)!}\right]$$

mais

$$\frac{C_k^j}{k+1-j} = \frac{C_{k+1}^j}{k+1}$$

$$Z_{k+2} = \frac{1}{\frac{N}{2}-k-1}\left[-Z_{k+1} + \frac{1}{(k+1)!}\sum_{j=0}^{k+1}(-1)^j\,C_{k+1}^j\right]$$

$$Z_{k+2} = -\frac{Z_{k+1}}{\frac{N}{2}-k-1} = (-1)^{k+1}\,\frac{(\frac{N}{2}-k-2)!}{\frac{N}{2}!}$$

Si k+2 = K

$$Z_K = (-1)^{K+1}\,\frac{(\frac{N}{2}-K)!}{(\frac{N}{2})!}$$

On a donc bien :

$$S = f(\frac{Ln(2)}{a}) + (-1)^{K+1}\,\frac{(\frac{N}{2}-K)!}{(\frac{N}{2})!}\,\alpha_K +$$

$$\sum_{i=1}^{K}\frac{(-1)^{i-1}}{K!}\,C_K^i\;i\left[\frac{\alpha_{K+1}}{(\frac{N}{2}+1-i)^2} + \frac{\alpha_{K+2}}{(\frac{N}{2}+1-i)^3} + \ldots\right]$$

On montre en majorant ce dernier terme que celui-ci est égal à :

$$o\left(\frac{(\frac{N}{2}-K)!}{(\frac{N}{2})!}\right).$$

Faisons $K = \frac{N}{2}$ et $a = \frac{Ln(2)}{t}$

Alors

$$f(t) = \sum_{i=1}^{N/2} x_i(\tfrac{N}{2})\;E_{\frac{N}{2}+i-1}(f).$$

Montrons que :

$$f(t) = \frac{Ln(2)}{t} \sum_{j=1}^{N} V_j \; F\left(j \frac{Ln(2)}{t}\right)$$

avec

$$V_j = (-1)^{N/2+j} \sum_{k=\left[\frac{j+1}{2}\right]}^{\min(j,\frac{N}{2})} \frac{k^{N/2} \; (2k)!}{(\frac{N}{2}-k)! \; k! \; (k-1)! \; (j-k)! \; (2k-j)!}$$

En effet :

$$f(t) = \sum_{k=1}^{N/2} x_k(\tfrac{N}{2}) \; E_{\frac{N}{2}+k-1}(f)$$

$$f(t) = \sum_{k=1}^{N/2} \frac{(-1)^{k-1} C_{N/2}^{k} k (\frac{N}{2}+1-k)^{N/2-1}}{(\frac{N}{2})!} \times \frac{Ln(2)}{t} \times \frac{(N+2-2k)!}{(\frac{N}{2}+1-k)! \; (\frac{N}{2}-k)!} \times$$

$$\sum_{i=0}^{N/2+1-k} C_{\frac{N}{2}+1-k}^{i} \; (-1)^{i} \; F\left((\tfrac{N}{2}+1-k+i) \frac{Ln(2)}{t}\right)$$

$$f(t) = \frac{Ln(2)}{t} \sum_{k=1}^{N/2} \sum_{i=0}^{\frac{N}{2}+1-k} \frac{(-1)^{i+k-1} (\frac{N}{2}+1-k)^{\frac{N}{2}-1} (N+2-2k)! \; F((\frac{N}{2}+1-k+i)\frac{Ln(2)}{t})}{(\frac{N}{2}-k)! \; (k-1)! \; (\frac{N}{2}-k)! \; i! \; (\frac{N}{2}+1-k-i)!}$$

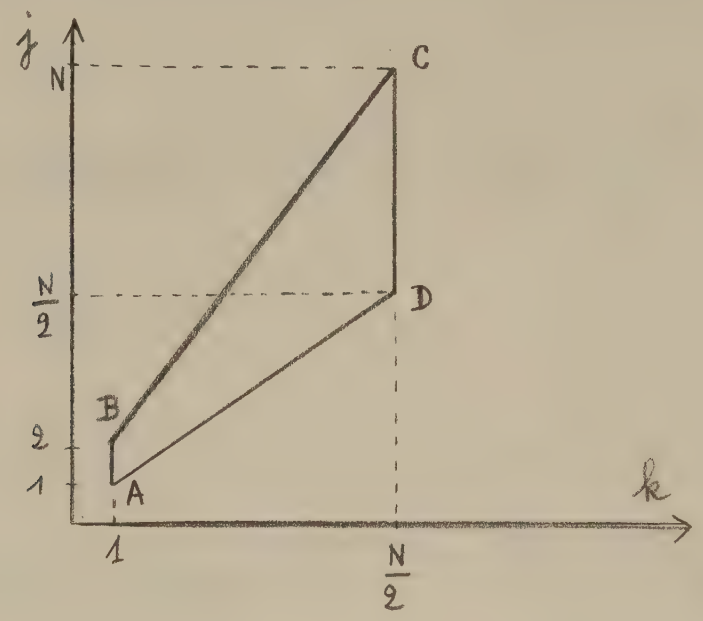
après avoir développé les coefficients du binôme et simplifié.

Posons alors $j = \frac{N}{2}+1-k+i$. Il vient :

$$f(t) = \frac{Ln(2)}{t} \sum_{k=1}^{N/2} \sum_{j=\frac{N}{2}+1-k}^{N+2-2k} \frac{(-1)^{j-\frac{N}{2}} (\frac{N}{2}+1-k)^{\frac{N}{2}-1} (N+2-2k)! \; F(j \frac{Ln(2)}{t})}{(\frac{N}{2}-k)! \; (k-1)! \; (\frac{N}{2}-k)! \; (j-\frac{N}{2}-1+k)! \; (N+2-2k-j)!}$$

Pour simplifier l'écriture nous changeons sous les signes Σ l'expression $(\frac{N}{2}+1-k)$ en k. Ce qui ne change rien puisque k varie de 1 à $\frac{N}{2}$.

Pour intervertir les sommations de façon à obtenir la forme cherchée, on peut faire le schéma suivant :



qui représente l'aire dans laquelle varient les indices.
On voit donc que si j varie de 1 à N alors k doit varier de Entier $(\frac{j+1}{2})$ pour être à droite de la droite BC jusqu'à $\min(j, \frac{N}{2})$ pour rester à gauche des lignes AD$(j < \frac{N}{2})$ et DC$(\frac{N}{2} < j)$.

Il vient alors :

$$f(t) = \frac{\mathrm{Ln}(2)}{T} \sum_{j=1}^{N} \sum_{K=[\frac{j+1}{2}]}^{\min(j,\frac{N}{2})} \frac{k^{\frac{N}{2}}\,(2k)!\;F(j\,\frac{\mathrm{Ln}(2)}{T})\;(-1)^{j-\frac{N}{2}}}{(\frac{N}{2}-k)!\,(k-1)!\,k!\,(j-k)!\,(2k-j)!}$$

V-2 - Procédure LINV et exemple d'activation

La méthode qui vient d'être théoriquement motivée a été programmée en ALGOL 60 par Stehfest sous forme de procédure appelée LINV. Nous en donnons ci-après la liste ainsi qu'un exemple d'activation.

Cette procédure à six paramètres :

P — réelle procédure a un paramètre formel réel exprimant la fonction F(p) dont on cherche l'originale (LINV n'utilise que des valeurs de F(p) où p est réel).

N — entier (voir V-3).

T — réel pour lequel on veut calculer f(T).

FA — réel qui contient après exécution de la procédure la valeur f(T) cherchée.

V — tableau réel vide à l'appel et qui contient après exécution de la procédure les éléments V_i (voir V-1).

M — entier. La procédure LINV calcule f(T) pour une valeur de T. En général elle est donc activée dans une boucle afin de tabuler f(T) pour plusieurs valeurs de T. Or pour une même fonction F(p) les V_i ne dépendent que de N. LINV ne calcule les V_i que si M est différent de N à l'appel et donne à M la valeur de N. Ainsi dans un appel itératif les V_i ne sont calculés qu'une fois. Ceci implique que le paramètre effectif correspondant à M est une variable et non une constante.

```
//LINV1 JOB (6053,0154,'F.VEILLON',MSGLEVEL=1
//EXEC  IMALGOL,PARM.ALGOL='LP,SIZE=45056,NS',TIME.GO=6
//ALGOL.SYSTEXT DD *
'BEGIN''PROCEDURE'LINV(P,N,T,FA,V,M);
     'VALUE'N,T;
     'INTEGER'M,N;'REAL'T,FA;'ARRAY'V;
     'REAL''PROCEDURE'P;
     'BEGIN''INTEGER'I,IH,K,NH,SN;
          'REAL'A;
          'ARRAY'G(/0:N/),H(/1:N/2/);
          'IF'M=N'THEN''GOTO'C;
          G(/0/):=1;
          NH:=N/2;
          'FOR'I:=1'STEP'1'UNTIL'N'DO'
               G(/I/):=G(/I-1/)*I;
          H(/1/):=2/G(/NH-1/);
          'FOR'I:=2'STEP'1'UNTIL'NH'DO'
               H(/I/):=1**NH*G(/2*I/)/(G/NH-I/)*G(/I/)*G(/I-1/));
          SN:=2*SIGN(NH-NH'/'2*2)-1;
          'FOR'I:=1'STEP'1'UNTIL'N'DO'
          'BEGIN'V(/I/):=0;
                'FOR'K:=(I+1)'/'2'STEP'1'UNTIL''IF'I<NH
                     'THEN'I'ELSE'NH'DO'
                     V(/I/):=V(/I/)+H(/K/)/(G(/I-K/)*G(/2*K-I/));
               V(/I/):=SN*V(/I/);
               SN:=-SN;
          'END';
          M:=N;
          C:FA:=0;A:=LN(2)/T;
          'FOR'I:=1'STEP'1'UNTIL'N'DO'
               FA:=FA+V(/I/)*P(I*A);
          FA:=A*FA;
     'END' LINV;
     'REAL''PROCEDURE'F9(P);'REAL'P;
     'BEGIN'
          F9:=2.0/P/(EXP(P)+EXP(-P));
     'END';
     'REAL''PROCEDURE'R9(T);
     'REAL'T;
     'BEGIN'
          R9:=1.0;
     'END';
     'REAL'T,FA,E,DELTAE,EM;
     'ARRAY'V(/1:10/);
     'INTEGER'M;
     M:=1;
     E:=0.1;DELTAE:=0.1;EM:=10.0;
     'FOR'T:=E'STEP'DELTAE'UNTIL'EM'DO'
     'BEGIN'
          LINV(F9,10,T,FA,V,M);
          OUTSTRING(1,'(' ')');
          OUTINTEGER(1,ENTIER(T));
          OUTREAL(1,FA);
          OUTREAL(1,R9(T));
          SYSACT(1,14,1);
     'END';
'END';
/*
```

V-3 - Considérations pratiques sur cette méthode

Théoriquement, plus N est grand, plus f(t) est précis. Pratiquement les erreurs d'arrondi provenant du calcul sur ordinateur deviennent très grandes quand N croît parce que les V_i prennent des valeurs absolues de plus en plus grandes.

Pour t et F(p) le nombre N pour lequel la précision est la plus grande croît avec le nombre de chiffres significatifs utilisés pour la représentation en machine des valeurs.

Plus $E_n(f)$ converge rapidement vers f(t), plus la valeur optimum de N est petite. Dans ce qui suit nous utilisons le terme 'sage' pour exprimer que la convergence de $E_n(f)$ vers f(t) est suffisamment rapide.

Une fonction oscillante $f(\tau)$ n'est pas 'sage' a moins que la période des oscillations ne soit grande par rapport à la moitié de la largeur du pic que $f_{\frac{N}{2}}(\frac{Ln(2)}{t},\tau)$ possède pour $\tau = t$.

Il ne faut pas s'attendre à des résultats très précis non plus si $f(\tau)$ a des discontinuités au voisinage de t. Si $f(\tau)$ est égale dans le voisinage de deux valeurs différentes de t, le résultat attaché à la plus petite de ces deux valeurs de t parce que le pic de $f_n(\frac{Ln(2)}{t},\tau)$ s'élargit à mesure que t croît. De nombreux essais ont été faits et ont montré que la valeur N optimum était proportionnelle à une constante près au nombre de chiffres significatifs de la machine utilisée.

Pour une machine à 8 chiffres décimaux, en simple précision nous avons pris N = 10 et en double précision N = 18.

Enfin Stehfest précise qu'il est imprudent de se reposer entièrement sur les résultats donnés par cette méthode si l'on n'est pas sûr que f(t) ne possède pas de discontinuités, de points de rebroussement, de pics pointus ou d'oscillations rapides et conseille de comparer les résultats avec ceux d'autres méthodes.

CHAPITRE - VI

METHODE DE DUBNER ABATE [DA]

Cette méthode est basée sur la relation qui existe entre la transformée de Laplace et la transformée de Fourier.

Soit une fonction f(t) réelle et F(p) sa transformée de Laplace. On peut écrire, c'est un résultat classique :

$$\mathrm{Re}[F(p)] = \int_0^\infty e^{-at} f(t) \cos bt \, dt \qquad (1)$$

$$f(t) = \frac{2e^{at}}{\pi} \int_0^\infty \mathrm{Re}[F(p)] \cos bt \, db \qquad (2)$$

avec $p = a + ib$.

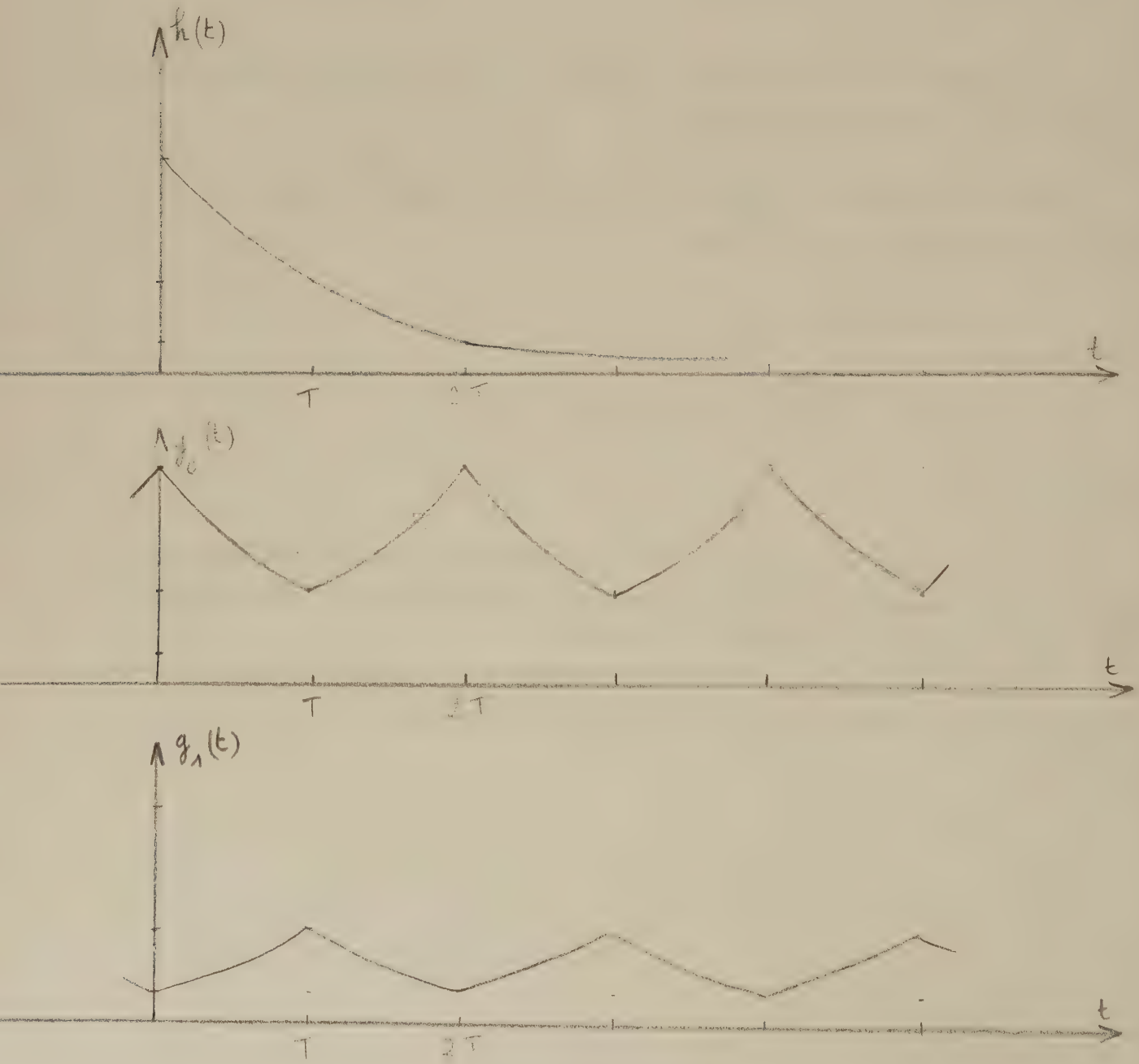
$a > 0$ arbitraire mais tel qu'il est plus grand que les parties réelles de toutes les singularités de F(p). Dans ce qui suit on suppose donc que F(p) n'a pas de singularités à droite de l'origine.

Dans le cas où elle en aurait, un changement de variable correspondant à une translation correcte de l'axe imaginaire permettrait d'appliquer encore ce qui suit.

VI-1 - Méthode

Soit une fonction h(t) quelconque mais nulle sur l'intervalle $]-\infty,0]$.

Considérons cette fonction sur des segments $[nT, (n+1)T]$ avec $n = 0,1,2,...$ et la suite infinie de fonctions paires $g_n(t)$ obtenue en reproduisant indéfiniment par symétrie (par rapport à un axe passant par une de ses bornes et perpendiculaire à l'axe des t) chacun de ces segments.



Il vient :

$$g_n(t) = \begin{cases} h(t) & nT \leq t \leq (n+1)T \\ h(2nT-t) & (n-1)T \leq t \leq nT \end{cases} \qquad (3)$$

avec n = 0,1,2,...

A seule fin de donner des $g_n(t)$ une représentation en série de Fourier, faisons le changement de variable de t en nT+t pour que les $g_n(t)$ soient définies sur le segment [-T,T].
Il vient :

$$g_n(t) = \begin{cases} h(nT+t) & 0 \leq t \leq T \\ h(nT-t) & -T \leq t \leq 0 \end{cases} \quad \text{pour n pair} \qquad (4)$$

et

$$g_n(t) = \begin{cases} h((n+1)T-t) & 0 \leq t \leq T \\ h((n+1)T+t) & -T \leq t \leq 0 \end{cases} \quad \text{pour n impair} \qquad (5)$$

La représentation de Fourier de chaque $g_n(t)$ est :

$$g_n(t) = \frac{A_{n,o}}{2} + \sum_{K=1}^{\infty} A_{n,k} \cos\left(\frac{k\pi t}{T}\right) \qquad (6)$$

où les coefficients qui sont les coefficients de Fourier ont la forme :

$$A_{n,k} = \frac{2}{T} \int_o^T h(nT+x) \cos\left(\frac{K\pi x}{T}\right) dx \quad \text{n pair} \qquad (7a)$$

$$A_{n,k} = \frac{2}{T} \int_o^T h((n+1)T-x) \cos\left(\frac{K\pi x}{T}\right) dx \quad \text{n impair} \qquad (7b)$$

Par simple changement de variable, on peut écrire plus simplement ces coefficients sous la forme :

$$A_{n,k} = \frac{2}{T} \int_{nT}^{T(n+1)} h(t) \cos\left(\frac{k\pi t}{T}\right) dt \qquad (7)$$

Alors

$$\sum_{n=0}^{\infty} g_n(t) = \frac{2}{T} \left[\frac{A(b_o)}{2} + \sum_{K=1}^{\infty} A(b_k) \cos\left(\frac{k\pi t}{T}\right)\right] \qquad (8)$$

où

$$A(b_k) = \int_{o}^{\infty} h(t) \cos\left(\frac{k\pi t}{T}\right) dt \qquad (9)$$

Si nous introduisons un facteur d'atténuation e^{-at} en posant :

$$h(t) = e^{-at} f(t) \qquad (10)$$

$A(b_k)$ est alors la transformée de Laplace d'une fonction réelle f(t). C'est-à-dire que :

$$A(b_k) = Re[F(p)] \text{ avec } p = a+i\frac{k\pi}{T}$$

L'équation 8 devient :

$$\sum_{n=0}^{\infty} e^{at} g_n(t) = \frac{2e^{at}}{T} \left[\frac{1}{2} Re[F(a)] + \sum_{K=1}^{\infty} Re\left[F\left(a+\frac{iK\pi}{T}\right)\right] \cos\frac{k\pi t}{T}\right] \qquad (11)$$

où nous avons multiplié les deux membres par e^{at}.

Le membre gauche de l'équation (11) est presque la transformée de Laplace F(p) sur l'intervalle [0,T] mais il contient une erreur.

Calculons ce membre gauche à l'aide des équations (4) et (5)

$$\sum_{n=0}^{\infty} e^{at} g_n(t) = \sum_{n=0}^{\infty} e^{at} h(2nT+t) + \sum_{n=1}^{\infty} e^{at} h(2nT-t) \qquad (12)$$

En se servant aussi de l'équation (10) il vient :

$$\sum_{n=0}^{\infty} e^{at} g_n(t) = f(t) + \text{erreur} \qquad (13)$$

$$\text{où erreur} = \sum_{n=1}^{\infty} e^{-2aTn} \left[f(2nT+t)+e^{2at} f(2nT-t)\right] \qquad (14)$$

Cette erreur devient très grande quant t est très peu différent de T. Dubner et Abate ont montré qu'en se restreignant au segment $[0, \frac{T}{2}]$ et en jouant sur le paramètre a, cette erreur peut être rendue suffisamment petite.

VI-2 - Conclusions

Sur l'intervalle $[0, \frac{T}{2}]$

$$f(t) \simeq \frac{2e^{at}}{T} \left[\frac{1}{2} \mathrm{Re}[F(a)] + \sum_{K=1}^{\infty} \mathrm{Re}[F(a+\frac{ik\pi}{T})] \cos \frac{k\pi t}{T}\right] \qquad (15)$$

En comparant les équations (15) et (2), il apparait que cette formule n'est autre qu'une formule du trapèze. Mais de Balbine et Franklin [ZBF] ont démontré que dans l'évaluation des intégrales de Fourier, la formule du trapèze est au moins aussi bonne que n'importe quelle règle de quadrature pour intégrales infinies.

Dans le cas présent cependant, l'erreur peut être rendue aussi petite que voulu en faisant varier le paramètre a, alors que dans la formule du trapèze l'erreur est directement liée au pas choisi.

VI-3 - APPROFONDISSEMENT SUR L'ERREUR

1°) Si t tend vers T le premier terme de la somme :

$$e^{-2aT} \left[f(2T+t)+e^{2at} f(2T-t)\right]$$

comporte un terme voisin de f(T). Ceci n'est pas acceptable évidemment et oblige à limiter l'intervalle sur lequel on peut calculer f(t) à l'intervalle $[0, \frac{T}{2}]$ par exemple.

Montrons que dans ces conditions l'erreur peut être rendue suffisamment petite.

a) Comme nous avons supposé que F(p) n'a pas de singularités à droite de l'origine, alors f(t) est bornée à partir d'une certaine abscisse t_o par une fonction de la forme Ct^m où C est une constante et m un entier non négatif.

Le premier terme de l'erreur devient pour $t = \frac{T}{2}$

$$E_1 = e^{-2aT} \left[f(\tfrac{5}{2} T) + e^{aT} f(\tfrac{3}{2} T) \right]$$

Si $\frac{3}{2} T \geq t_o$ il vient :

$$E_1 \leq e^{-2aT} \left[C(\tfrac{5}{2} T)^m + e^{aT} C(\tfrac{3}{2} T)^m \right]$$

$$E_1 \leq e^{-aT} C(\tfrac{3}{2} T)^m + e^{-2aT} C(\tfrac{5}{2} T)^m$$

Il faut donc que aT soit tel que $e^{-aT} C(\frac{3}{2} T)^m$ soit petit et que $e^{-2aT} C(\frac{5}{2} T)^m$ soit petit devant $e^{-aT} C(\frac{3}{2} T)^m$.
On est assuré que les termes suivants de la somme seront négligeables puisque modulés par e^{-2aTn}.

L'erreur est donc de l'ordre de $C(1,5T)^m e^{-aT}$.
Ceci présente le grave inconvénient de faire intervenir la connaissance du comportement de la solution.

Remarque :

Dans le cas où f(t) est inférieure ou égale à une constante C sur l'intervalle $[0,\infty]$ l'erreur est sommable (c'est une progression géométrique) et vaut pour $t = \frac{T}{2}$

$$\text{erreur} \leq C\, e^{-aT} \frac{(1+e^{-aT})}{1-e^{-2aT}}.$$

CHAPITRE - VII

ETUDE DETAILLE ET AMELIORATIONS APPORTEES A LA METHODE DE DUBNER ABATE

VII-1 Etude de l'erreur

Rappelons que cette méthode se ramène simplement au calcul de l'expression :

$$\tilde{f}(t) = \frac{2e^{at}}{T}\left[\frac{1}{2}\,\mathrm{Re}[F(a)+\sum_{k=1}^{\infty}\mathrm{Re}[F(a+\frac{ik\pi}{T})]\cos\frac{k\pi t}{T}\right]$$

qui est une valeur approchée de f(t) telle que :

$$f(t) = \tilde{f}(t) + err$$

et

$$Err = \sum_{n=1}^{\infty} e^{-2aTn}\,[f(2nT+t)+e^{2at}\,f(2nT-t)]$$

Appelons $f_c(t)$ la valeur calculée de f(t).
On peut écrire :

$$f(t) = f_c(t) + Err_c + Err$$

où Err_c est l'erreur de calcul sur l'expression.

a) Remarque sur Err

Nous avons vu que cette erreur peut être rendue suffisamment petite en jouant sur le paramètre a. Mais ceci implique que l'on connaisse l'ordre de grandeur de la croissance de f(t), ce qui n'est en général pas le cas.

Il peut être intéressant de calculer pour avoir une approximation de Err la somme des 3 ou 4 premiers termes de la suite (elle décroît très vite) en remplaçant f par f_c. Soit :

$$\tilde{Err} = \sum_{n=1}^{3 \text{ ou } 4} e^{-2aTn}\left[f_c(2nT+t)+e^{2at} f(2nT-t)\right]$$

b) Remarques sur Err_c

Err_c est l'erreur que l'on fait en calculant la somme infinie qui intervient dans $\tilde{f}(t)$. Or $\tilde{f}(t)$ est d'autant moins dépendante du paramètre a qu'elle est mieux calculée. C'est pourquoi nous avons cherché à raffiner au maximum le calcul de cette somme infinie.

VII-2 - Méthode d'accélération de la convergence de la somme

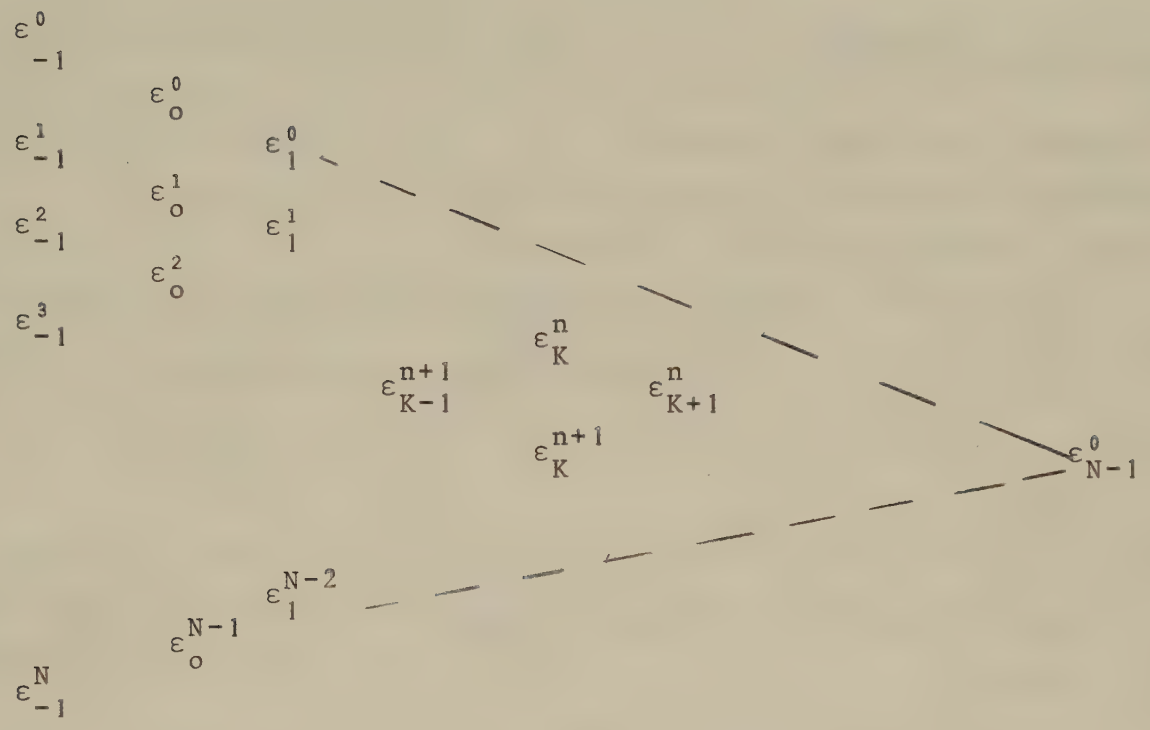
ε-algorithme [WY1] [WY2] [BR]

a) Principe de l'ε-algorithme

Etant donnés une suite de terme général U_n, S_n le terme général de la série associée, S sa limite, l'ε-algorithme permet de calculer itérativement à l'aide des N premiers termes de la suite $S_o,\ldots,S_{N-1}$ une valeur S' qui est une meilleure approximation de S que S_N.

b) Mise en oeuvre

L'algorithme permet de construire de proche en proche, colonne par colonne, le tableau triangulaire suivant :



où

$$\varepsilon_{K+1}^{n} = \varepsilon_{K-1}^{n+1} + \frac{1}{\varepsilon_{K}^{n+1} - \varepsilon_{K}^{n}} \qquad \text{pour} \quad \begin{cases} n = 0,1,\ldots,N-2 \\ k = 0,1,\ldots,N-1 \end{cases}$$

$$\varepsilon_{-1}^{n} = 0 \qquad \text{pour} \quad n = 0,1,\ldots,N$$

$$\varepsilon_{o}^{n} = S_n \qquad \text{pour} \quad n = 0,1,\ldots,N-1$$

Remarque

La première colonne qui est nulle n'est là que pour permettre la mise en marche de l'itération.

Les derniers éléments des colonnes de rang pair en numérotant celles-ci à partir de la gauche sont des approximations de S.

c) La procédure EPSALGOR

```
'PROCEDURE' EPSALGOR(A,N,RESULTAT) ;
'TABLEAU' A; 'ENTIER' N; 'TABLEAU' RESULTAT;
'DEBUT' 'TABLEAU' B(/1:N-1/); 'ENTIER' I,J,K;
    'FOR'I:=1'STEP'1'UNTIL'N'/'2'DO'RESULTAT(/I/):=0.0;
    'POUR' I:=1 'PAS' 1 'JUSQUA' N-1 'FAIRE'
   'DEBUT''IF'A(/I+1/)=A(/I/)'THEN'OUTSTRING(1,'('TERMES EGAUX')');
          B(/I/):=1/(A(/I+1/)-A(/I/));
     'END';
     RESULTAT(/1/) := A(/N/);
      K:=2;
     'POUR' J:=N-2 'PAS' -1 'JUSQUA' 1 'FAIRE'
     'DEBUT'
          'POUR' I:=1 'PAS' 1 'JUSQUA' J 'FAIRE'
          'DEBUT' A(/I/):=B(/I/);
               'IF'B(/I+1/)=B(/I/)'THEN''GOTO'FIN;
               B(/I/):=A(/I+1/)+1/(B(/I+1/)-B(/I/));
          'FIN';
          'SI' (K'/'2)*2 = K 'ALORS' RESULTAT(/(K'/'2)+1/) :=B(/J/);
           K:=K+1;
     'FIN';
FIN: 'FIN' EPSALGOR;
```

Cette procédure a pour paramètre A, N, et RESULTAT.

- A est un tableau tel que :

$$A[1] = S_0$$
$$\text{-----}$$
$$A[N] = S_{N-1}$$

- N est la dimension du tableau A
- RESULTAT contient les valeurs qui sont des approximations de la somme. Il a pour dimensions [1: N ÷ 2] si N est pair et [1 : N ÷ 2-1] si N est impair :

$$\text{RESULTAT}[1] = \varepsilon_o^{N-1} = S_N$$

$$\text{RESULTAT}[2] = \varepsilon_2^{N-3}$$

---- etc..

d) Conditions favorables d'application de l'ε-algorithme

On dit qu'une série a une convergence d'ordre p si :

$$\frac{S_{n+1}-S}{(S_n-S)^p} \to \ell \neq 0$$

L'expérience à montré que l'ε-algorithme donne de bons résultats si la convergence de la série est d'ordre 1. Mais ceci n'est pas facile à exploiter puisque S est justement ce que l'on cherche.

Brezinsky a montré que si S_n pouvait se mettre sous la forme :

$$S_n = S+\varphi(n,a) \text{ avec}$$

$$\varphi(n,a) = \sum_{i=1}^{p} A_i(n) r_i^n + \sum_{i=p+1}^{q} [B_i(n) \cos b_i n + C_i(n) \sin b_i n)] e^{s_i n} + \sum_{i=0}^{m} c_i \delta_{in}$$

$\varphi(n,a) \to 0$ quand $n \to \infty$,

A_i, B_i et C_i étant des polynômes en n tels que si d_i est égal au degré de A_i plus 1 pour i = 1,...,p et au plus grand des degrés de B_i et de C_i plus 1 pour i = p+1,...,q on ait :

$$m+1 + \sum_{i=1}^{p} d_i + 2 \sum_{i=p+1}^{q} d_i = k$$

avec la convention que m = -1 s'il n'y a pas de terme δ_{in} alors $\varepsilon_{2k}^n = S$. C'est-à-dire que l'ε-algorithme donne la valeur exacte de la somme cherchée.

La valeur a qui intervient dans $\varphi(n,a)$ est un vecteur formé de toutes les inconnues r_i, s_i, b_i, c_i, ainsi que les coefficients des polynômes A_i, B_i et C_i.

Cas particulier simple

$$S_n = S + \sum_{i=1}^{k} \alpha_i \lambda_i^n \quad \text{avec} \quad \alpha_i \neq 0 \quad |\lambda_i| < 1$$

L'ε-algorithme dans ce cas donne le résultat exact et pour cela il suffit de l'appliquer aux 2k+1 premiers termes de la série puisque ceux-ci suffisent pour calculer ε_{2k}^n.

Ces conditions sont évidemment rarement réalisées.
Il peut être intéressant de chercher si en groupant de certaines façons les termes de la suite dont on veut calculer la somme on n'obtient pas une suite équivalente (de même somme) qui serait plus proche des bonnes conditions d'utilisation de l'ε-algorithme. Nous verrons plus loin un essai sur un cas simple.

VII-3 - Etude du cas simple f(t) = 1

La fonction f(t) = 1 a pour transformée :

$$F(p) = \frac{1}{p}$$

et

$$\text{Re } F(p) = \frac{a}{a^2+b^2} \quad \text{avec } p = a+ib$$

Si nous reportons dans l'expression de $\tilde{f}(t)$:

$$\tilde{f}(t) = \frac{2e^{at}}{T} \left[\frac{1}{2a} + \frac{1}{a} \sum_{K=1}^{\infty} \frac{1}{1+\frac{k^2\pi^2}{T^2a^2}} \cos \frac{k\pi t}{T}\right]$$

Dans le calcul nous limitons la sommation a une valeur N de l'indice. Isolons le terme négligé

$$\tilde{f}(t) = \frac{2e^{at}}{T}\left[\frac{1}{2a} + \frac{1}{a}\sum_{K=1}^{N} \frac{1}{1+\frac{k^2\pi^2}{T^2a^2}} \cos\frac{k\pi t}{T}\right]$$

$$+ \underbrace{\frac{2e^{at}}{aT}\sum_{k=N+1}^{\infty} \frac{1}{1+\frac{k^2\pi^2}{T^2a^2}} \cos\frac{k\pi t}{T}}_{\text{terme négligé}}$$

a) Relation entre les paramètres a, T et la variable t

Nous avons déjà vu que T devait être, pour que la formule approchée garde un sens, au moins double de la plus grande valeur de t pour laquelle on veut calculer f(t). Or le calcul de $f(t_i)$ est indépendant du calcul de $f(t_{i+1})$. Il a donc paru naturel non pas de fixer T à partir de la valeur t_{max}, plus grande valeur de t pour laquelle on veut tabuler f(t) mais de fixer une fois pour toutes le rapport $\frac{T}{t} = \lambda_1$ avec $\lambda_1 \geq 2$.

Nous avons vu aussi que l'erreur de principe Err était fonction de e^{-aT}.
Prenons $aT = \lambda_2$.
Il vient alors :

$$at = \frac{\lambda_2}{\lambda_1} = \lambda$$

On voit qu'avec ces conventions le terme négligé devient constant.
Ceci n'est évidemment pas toujours vrai et dépend de la fonction traitée.
Mais il y a tout de même deux intérêts à mener les calculs de cette façon.

1°) L'expérience a montré que même pour les fonctions telles que le terme négligé n'est pas rendu constant par ces conventions les calculs se trouvent améliorés.

2°) Si on sait que l'erreur est constante, et si par exemple on connait f(t) pour une valeur t_o de t on pourra en tirer le terme correctif.

Remarque - Les fonctions $Re[F(a+ib)]$ telles que le terme négligé devient constant si on prend pour mener les calculs $\frac{T}{t} = \lambda_1$ et $aT = \lambda_2$ sont les fonctions qui peuvent se mettre sous la forme :

$$Re[F(a+ib)] = \frac{1}{a} G(\frac{b}{a})$$

Posons $a = u$ et $b/a = v$

Une condition nécessaire et suffisante pour que $Re[F]$ puisse se mettre sous cette forme est que :

$$\frac{\frac{\partial Re[F]}{\partial u}}{Re[F]} = A(u)$$

$$\frac{\frac{\partial Re[F]}{\partial v}}{Re[F]} = B(v)$$

Mais il est aussi simple souvent de vérifier directement.

b) Influence du nombre N auquel on arrête la sommation

Nous venons de voir qu'une certaine façon de lier les paramètres entre eux favorise les calculs. Ces conventions ont une conséquence sur le terme en cosinus qui apparait dans le terme de la somme.

Ce terme est :

$$\cos \frac{k\pi t}{T}$$

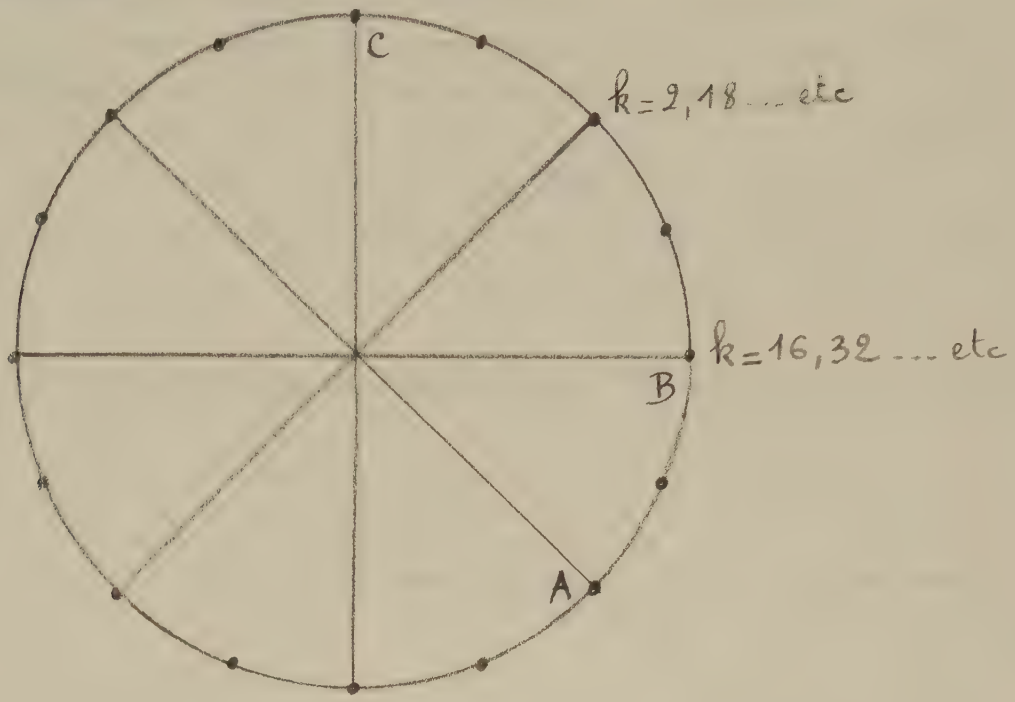
Soit : $\cos \frac{k\pi}{\lambda_1}$

Ceci implique que :

$$Re[F(a+ \frac{ik\pi}{T})] \quad \text{et} \quad Re[F(a+ \frac{i(k+2\lambda_1)\pi}{T})]$$

sont affectés du même cosinus.

Prenons comme exemple $\lambda_1 = 8$. On voit alors facilement sur le schéma suivant comment sont distribués les facteurs en cosinus



Nous nous sommes demandés s'il n'y avait pas une valeur optimale de N modulo 16 pour arrêter la sommation. Nous avons fait des essais avec N = 30, 32, 36 correspoint aux points A, B, C du schéma précédent.

Il est apparu que la meilleure solution était la solution C (ou sa symétrique par rapport à O), la seconde en qualité la solution A, la moins bonne la solution B. Ceci veut dire que plus le facteur en cosinus du terme sur lequel on arrête la sommation est petit, meilleur est le résultat. Ceci vient du fait qu'on ne prend pas S_N comme valeur de la somme mais qu'on applique l'ε-algorithme sur les N premiers termes $S_0,\ldots,S_N$ et que l'ε-algorithme est sensible à la valeur du dernier terme S_N.

Les essais ont été faits sur un cas assez simple :

$$f(t) = \frac{1}{\sqrt{\pi\, t}} \quad \text{dont la transformée est}$$

$$F(p) = \frac{1}{\sqrt{p}}$$

Nous allons voir en c) pourquoi nous n'avons toutefois pas retenu cette façon de faire.

c) Blocage des termes de la somme

Si la convergence de l'ε-algorithme est lente et que l'on est amené à le mettre en oeuvre sur beaucoup de termes, les calculs peuvent devenir très lourds. Nous avons pensé à transformer la suite de la façon suivante :

Soit

$$U_k = \text{Re } F\left(a + \frac{ik\pi}{T}\right)$$

$$V_k = \sum_{\ell=1}^{k} \text{Re } F\left(a + \frac{i\ell\pi}{T}\right)$$

Au lieu d'appliquer l'ε-algorithme sur les N termes :

$$V_1, \ldots, V_N$$

nous avons essayé de l'appliquer sur les $\frac{N}{\lambda_1}$ termes :

$$W_1, \ldots, \frac{W_N}{\lambda_1}$$

(il faut alors N multiple de λ_1).

tels que :

$$W_1 = \sum_{m=1}^{\lambda_1} V_m$$

$$\cdots$$

$$W_p = \sum_{m=\lambda_1(p-1)+1}^{\lambda_1 p} V_m$$

Ceci revient à avoir groupé les termes consécutifs de la suite dont les facteurs en cosinus étaient tels que leurs arcs étaient sur un même demi-cercle.

Les résultats sont à peu près les mêmes et les calculs nettement diminués. L'ε-algorithme appliqué à N termes représente le calcul de $\frac{N(N+1)}{2}$ termes ε_i^j (4 références à des éléments de tableau, une addition, une soustraction, une division). En groupant les termes comme nous venons de le montrer on ramène le nombre d'ε_i^j calculés à :

$$\frac{\frac{N}{\lambda_1}\left(\frac{N}{\lambda_1}+1\right)}{2}.$$

Noter que cette façon de faire inhibe ce qui a été dit au §. b) puisque N doit être un multiple de λ_1.

d) Valeur optimale de λ_1

Sur le cas simple f(t) = 1 nous avons essayé $\lambda_1 = 4, 8, 12$. Il a résulté de ces essais que $\lambda_1 = 8$ était la meilleure solution car c'est dans ce cas que l'erreur relative est la plus petite, pour la même valeur des autres paramètres.

Tableau des résultats de cette expérience

λ_1	erreur relative
4	$4.08 \quad 10^{-4}$
8	$5.05 \quad 10^{-7}$
12	$1.01 \quad 10^{-4}$

Les calculs ont été faits pour t variant de 1 à 10 par pas de 1 et de 20 à 100 par pas de 10.
Comme la fonction f(t) est constante et que l'on a vu que l'erreur était constante dans ce cas, il est naturel de trouver une valeur calculée de f(t) constante. L'expérience a vérifié ce fait.
Le programme correspondant se trouve en annexe car il comprend des améliorations non encore étudiées.
Les résultats sont donc, quel que soit t :

λ_1	f(t) calculée
4	1.000408
8	1.000000505
12	1.000101

VII-4 - Recherche de l'optimum du paramètre a

Nous avons vu que l'expression $\tilde{f}(t)$ que nous calculons est en principe indépendante de a mais en est en fait dépendante du fait :

a) que nous négligeons l'erreur de principe Err

b) que l'on fait une erreur de calcul Err_c.

Puisque $\tilde{f}(t)$ dépend de a, appelons la $\tilde{f}(t,a)$.
Nous avons essayé de chercher une valeur optimale de a de la façon suivante :
Pour une valeur de t fixée, nous calculons

$$\tilde{f}(t,a_i) \text{ pour n valeur de i}$$

Nous obtenons ainsi une description par points d'une courbe $\tilde{f}(t,a)$ par rapport à a.
$\tilde{f}(t,a)$ est indépendante ou peu dépendante de a aux points où sa dérivée $\tilde{f}_a(t,a)$ est nulle (s'il y en a) ou petite par rapport à a. Si cette dérivée s'annule plusieurs fois, c'est de toute façon la plus petite des valeurs de a ainsi déterminées qu'il convient de choisir car c'est elle qui minimise Err.

α) Algorithme de détermination du a optimal

Par les n points calculés déterminés par le couple :

$$\begin{cases} a_i \\ \tilde{f}(t,a_i) \end{cases}$$

nous faisons passer une fonction spline d'ordre deux . Pour chaque intervalle $[a_i,a_{i+1}]$ (en commençant par celui qui est le plus proche de l'origine) nous cherchons :

1°) si la dérivée de la fonction spline s'annule pour une valeur a_{dn}

$$a_i \leq a_{dn} \leq a_{i+1}$$

dans ce cas a_{dn} est le a optimal et c'est fini.

2°) sinon, la valeur $a_{dp,i}$ pour laquelle $\tilde{f}'_a(t,a)$ est la plus petite en valeur absolue.

Si la dérivée ne s'est jamais annulée on prend alors pour a optimal

$$\min_{1\leq i\leq n} (a_{dp,i})$$

β) Résultats

Pour la fonction $\frac{L_3(t)}{6}$ dont la transformée est

$$F_6(p) = \frac{(p-1)^3}{p^4}$$

et pour t = 1 on obtient :

$a_o = 1.15$ -0,666696

$a_1 = 1.20$ -0,666679

$a_2 = 1.25$ -0,666670

$a_3 = 1.30$ -0,666664

$a_4 = 1.35$ -0,666656

L'algorithme exposé ci-dessus montre que la dérivée de la spline d'interpolation s'annule pour

$$a_{dn} = 1,2706$$

et que : $\tilde{f}(1, a_{dn}) = -0,666667$

Soit une erreur relative de $1.08\ 10^{-6}$, la valeur vraie étant $-\frac{2}{3}$.

γ) Organigramme de la recherche du a optimal

γ1) Procédure de calcul de la fonction spline d'interpolation

Etant donnés (N+1) points p_i dont les abscisses x_i et les ordonnées y_i respectivement sont dans les éléments de tableau X[I] et Y[I], la procédure calcule dans les éléments de tableau M[I] les valeurs m_i de la dérivée seconde de la fonction spline aux points p_i.

Dans chaque intervalle $[p_i, p_{i+1}]$ la fonction spline est un polynôme de degré 3.

$$P = Ax^3 + Bx^2 + Cx + D$$

$$P' = 3Ax^2 + 2Bx + C$$

$$P'' = 6Ax + 2B$$

On a donc

$$m_i = 6Ax_i + 2B$$

$$m_{i+1} = 6Ax_{i+1} + 2B$$

Soit

$$A = \frac{m_{i+1} - m_i}{6(x_{i+1} - x_i)}$$

$$B = \frac{m_i - 6Ax_i}{2}$$

$$C = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - A(x_{i+1}^2 + x_i^2 + x_i x_{i+1}) - B(x_i + x_{i+1})$$

2) Organigramme

X[I] abscisses des points de la courbe (ce sont les a_i)

Y[I] ordonnées des points de la courbe (ce sont les $f(t,a_i)$)

X1, X2 racines réelles de la dérivée première de la fonction spline.

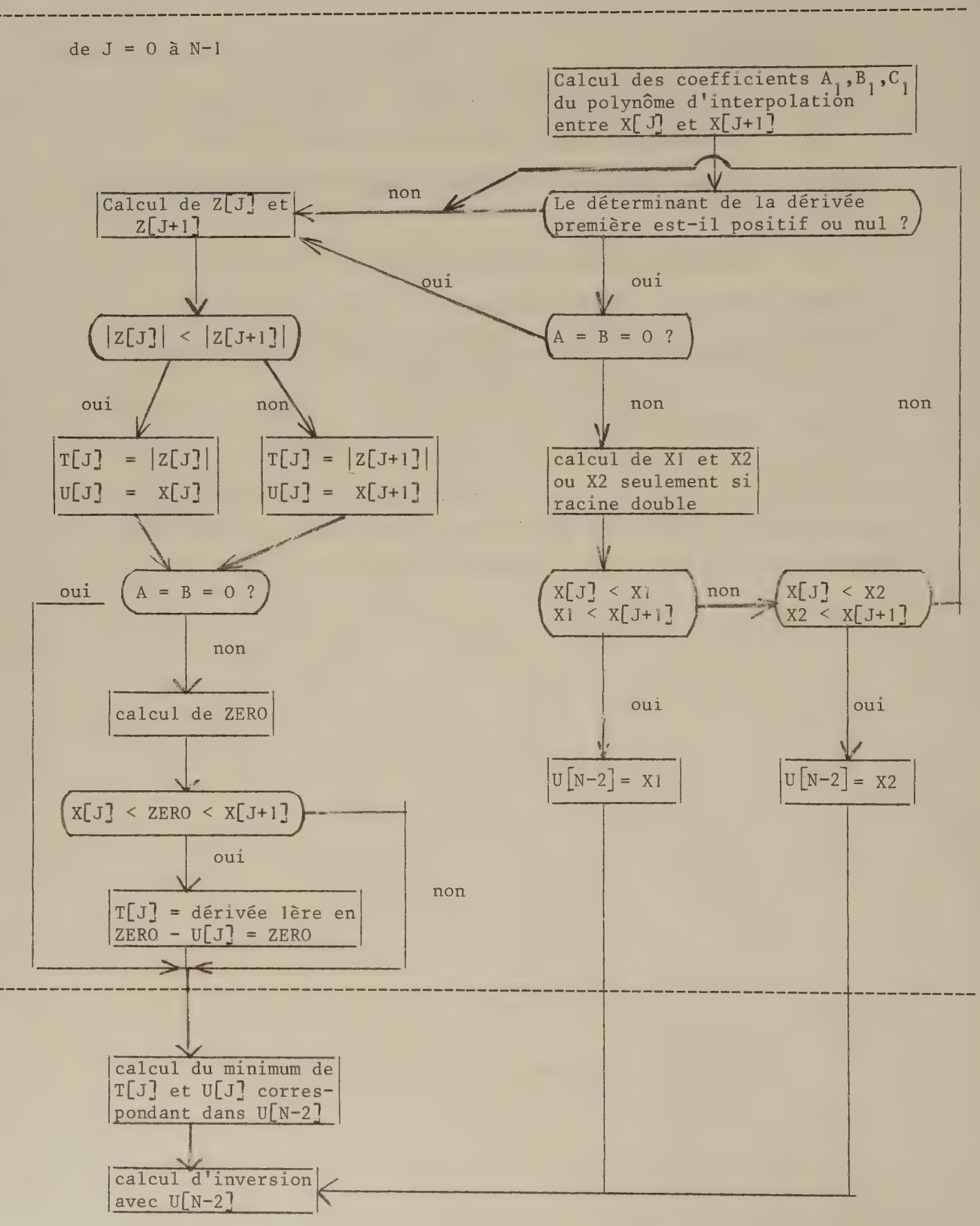
Z[I] dérivée première de la spline en x_i

ZERO zéro de la dérivée seconde

T[I] minimum de la valeur absolue de la dérivée première de la spline sur le segment $[x_i,x_{i+1}]$

U[I] abscisse correspondant à T[I].

En fin de travail on range dans U[N-2] soit l'abscisse du zéro de la dérivée première de la spline, soit l'abscisse du point de plus petite dérivée seconde en valeur absolue sur le segment $[x_o,x_n]$


de J = 0 à N-1
Calcul des coefficients A1,B1,C1 du polynôme d'interpolation entre X[J] et X[J+1]
Le déterminant de la dérivée première est-il positif ou nul ?
non
Calcul de Z[J] et Z[J+1]
oui
oui
A = B = 0 ?
|Z[J]| < |Z[J+1]|
oui
non
non
non
T[J] = |Z[J]| U[J] = X[J]
T[J] = |Z[J+1]| U[J] = X[J+1]
calcul de X1 et X2 ou X2 seulement si racine double
oui
A = B = 0 ?
X[J] < X1 X1 < X[J+1]
non
X[J] < X2 X2 < X[J+1]
non
calcul de ZERO
oui
oui
U[N-2] = X1
U[N-2] = X2
X[J] < ZERO < X[J+1]
oui
non
T[J] = dérivée 1ère en ZERO - U[J] = ZERO
calcul du minimum de T[J] et U[J] correspondant dans U[N-2]
calcul d'inversion avec U[N-2]

CHAPITRE - VIII

RESULTATS COMPARES DES DIFFERENTES METHODES

Les mérites respectifs des méthodes de Gaver Stehfest et de Dubner Abate ont été étudiés sur les exemples suivants :

$$f_1(t) = \frac{1}{\sqrt{\pi t}} \quad <--> \quad F_1(p) = \frac{1}{\sqrt{p}}$$

$$f_2(t) = -c - Ln(t) \quad <--> \quad F_2(p) \quad \frac{Ln(p)}{p} \qquad (1)$$

$$f_3(t) = \frac{t^3}{6} \quad <--> \quad F_3(p) = \frac{1}{p^4}$$

$$f_4(t) = e^{-t}\sqrt{t} + t(1-e^{-t}) \quad <--> \quad F_4(p) = \frac{\sqrt{\pi}}{2} \frac{1}{(p+1)^{3/2}} + \frac{1}{p^2} - \frac{1}{(p+1)^2}$$

$$f_5(t) = \frac{L_3(t)}{6} \quad <--> \quad F_5(p) = \frac{(p-1)^3}{p^4} \qquad (2)$$

$$f_6(t) = e^{-t} \quad <--> \quad F_6(p) = \frac{1}{p+1}$$

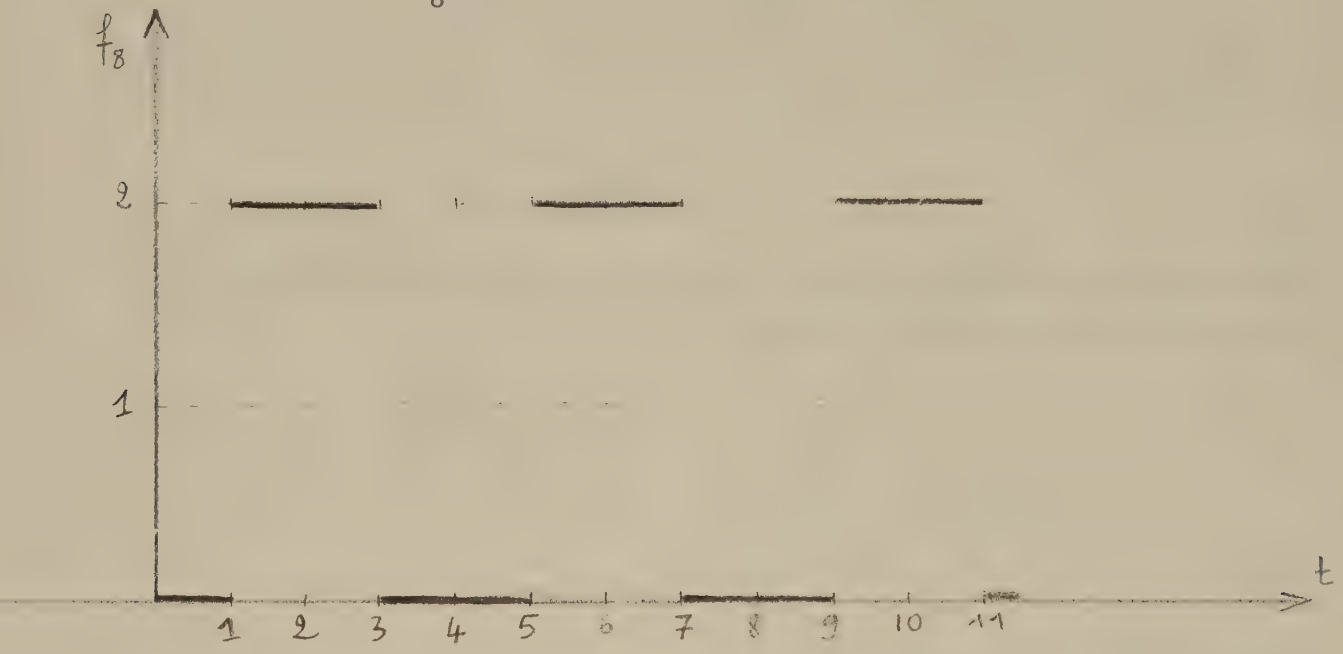
$$f_7(t) = 1 \quad <--> \quad F_7(p) = \frac{1}{p}$$

$$f_8(t) = \begin{cases} 0 \text{ si } \frac{1}{2} E\left(\frac{t+1}{2}\right) = E\left(\frac{t+1}{4}\right) \\ = 2 \text{ sinon} \end{cases} \quad <-->F_8(p) = \frac{1}{p \text{ ch } p} \qquad (3)$$

(1) $C = 0,5772157$

(2) $L_3(t) = -t^3+9t^2-18t+6$

(3) Le graphe de $f_8(t)$ est le suivant :

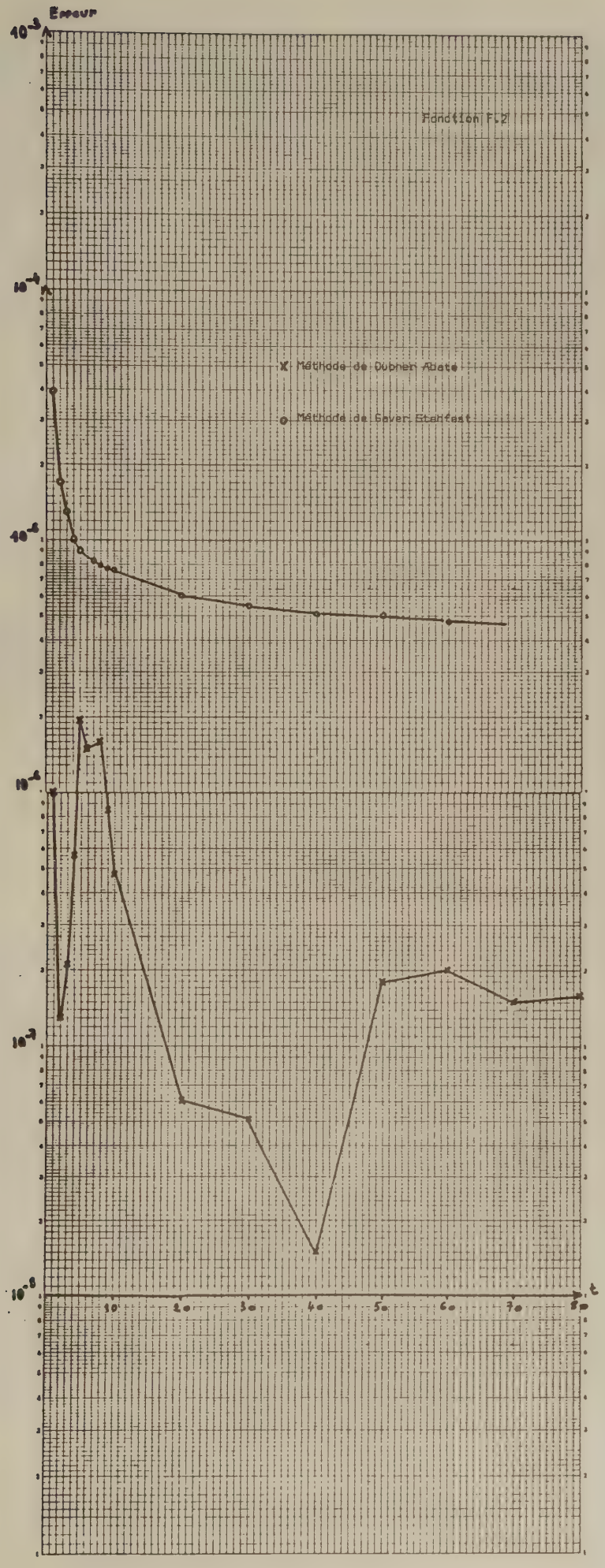


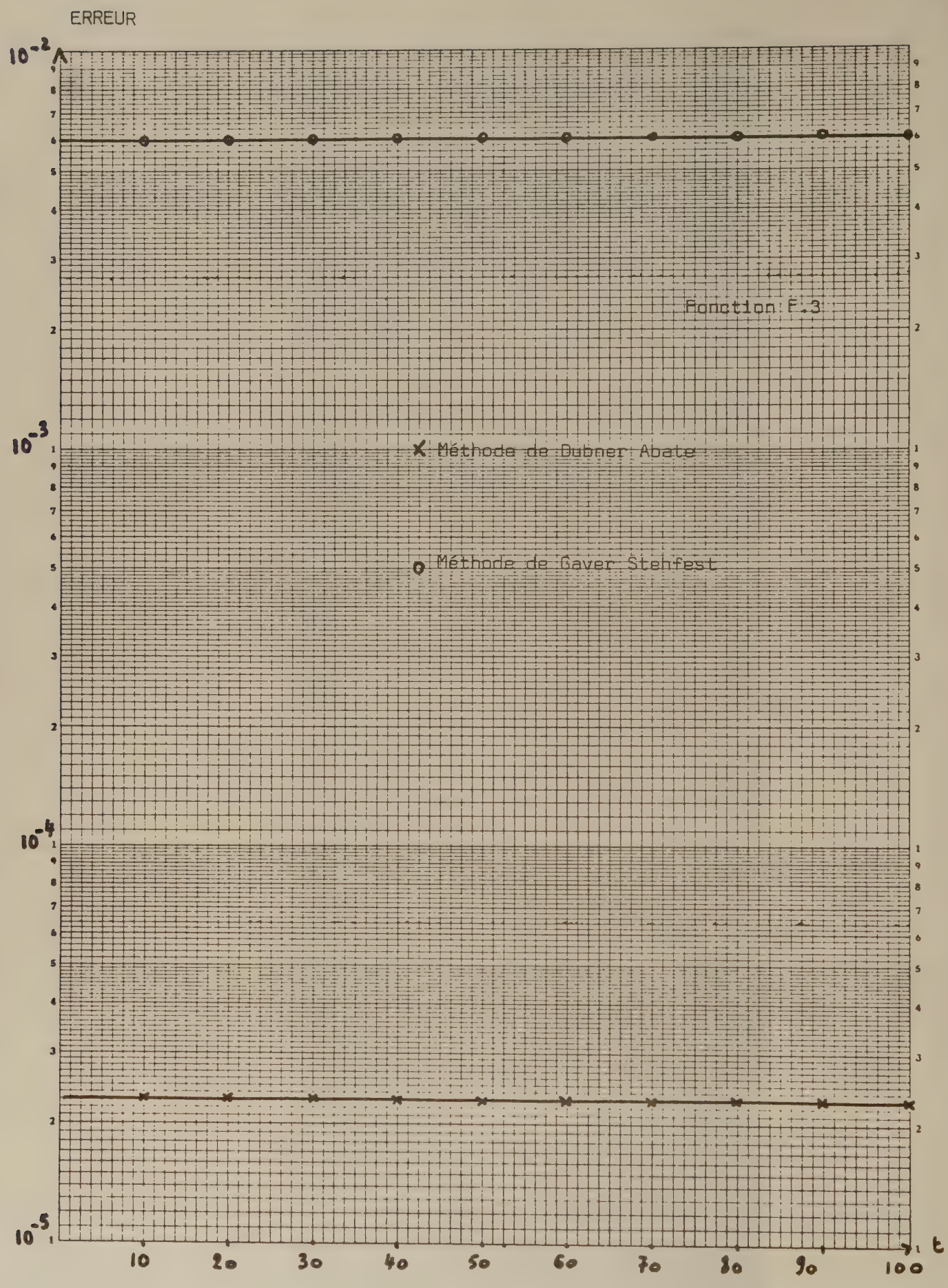
Nous avons représenté les résultats sous forme de courbes d'erreur.

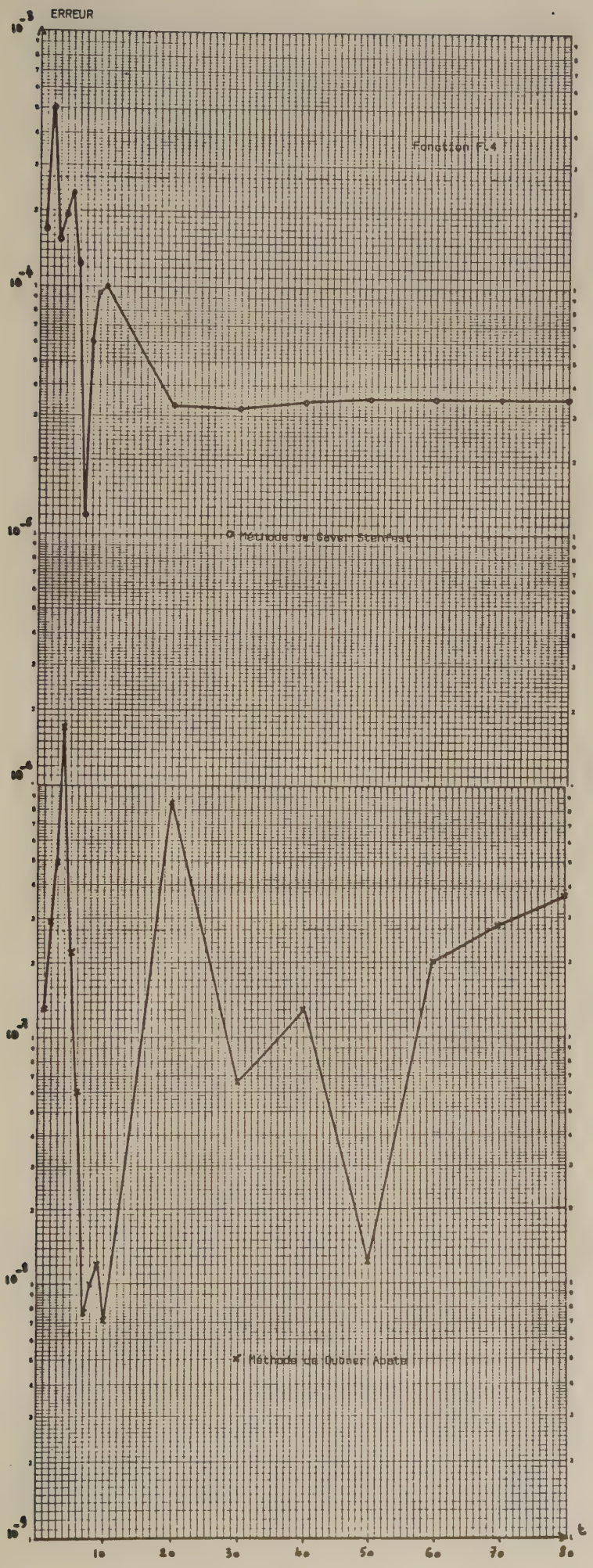
Remarque sur les résultats obtenus avec f_1

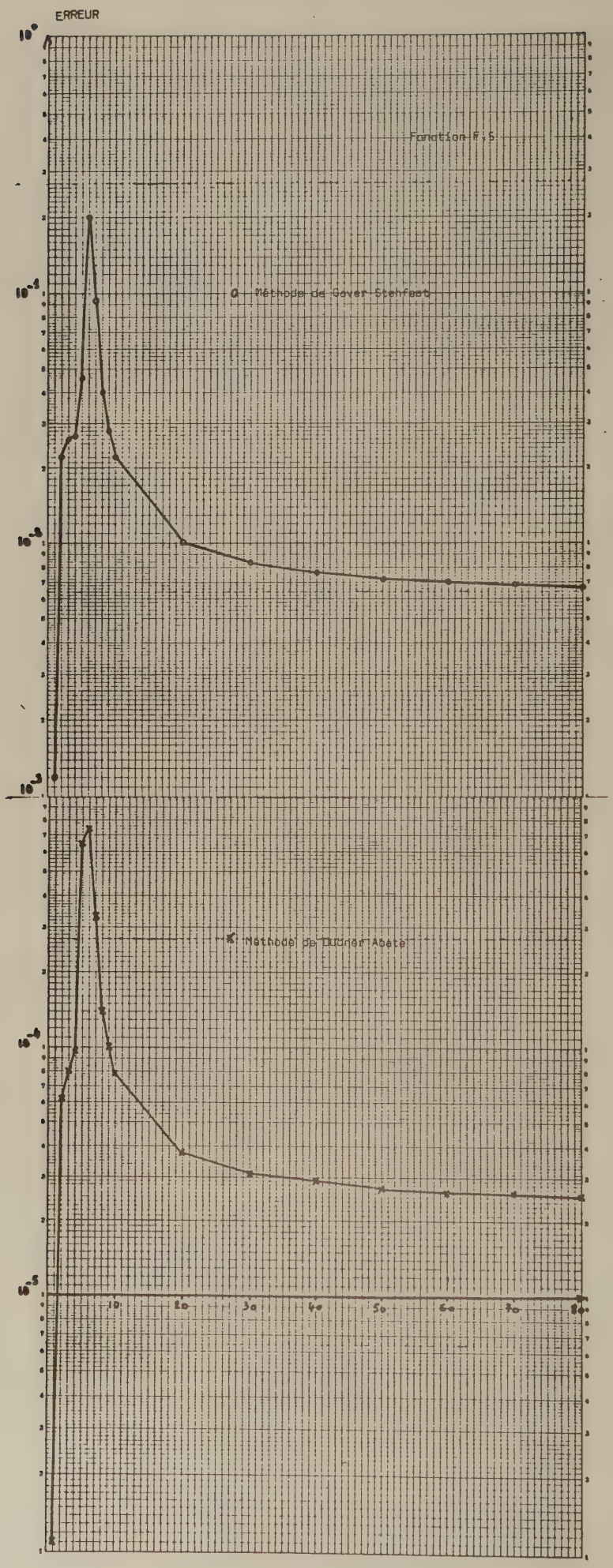
La fonction est telle que nous attendions à une erreur constante (VII.3). C'est le calcul du a optimal qui introduit ces légères variations.

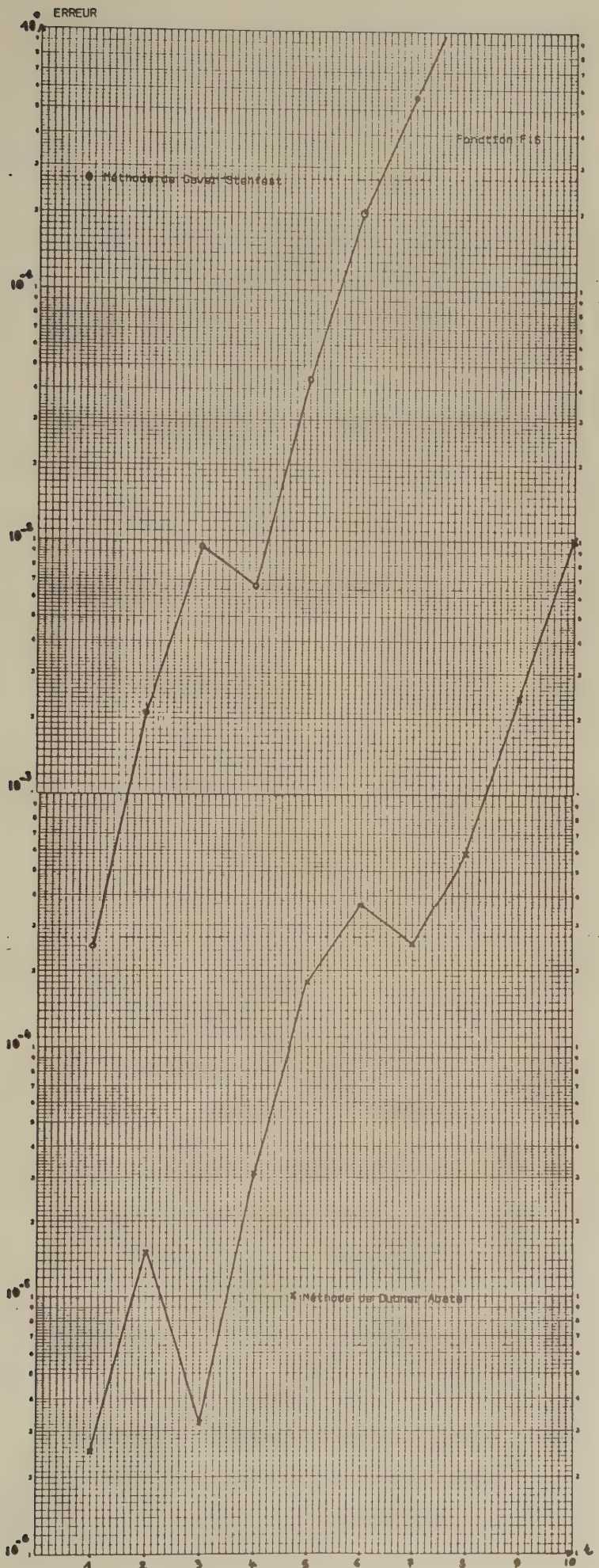
D'ailleurs, nous avions obtenu de meilleurs résultats (erreur constante et égale à $4.0.10^{-7}$) avec un autre ensemble de a_i ($a_1 = \frac{0,75}{t}$ ... $a_5 = \frac{0,95}{t}$) mais nous avons voulu garder l'homogénéité des paramètres utilisés, c'est pourquoi nous avons tout de même donné ces résultats un peu moins bons.


Erreur
Fonction F(2)
X Méthode de Dubner Abate
o Méthode de Saver Stehfest
t


ERREUR
10^{-2}
10^{-3}
10^{-4}
10^{-5}
Fonction F.3
Méthode de Dubner Abate
Méthode de Gaver Stehfest
10
20
30
40
50
60
70
80
90
100
t


ERREUR
Fonction F.4
o Méthode de Gaver Stehfest
x Méthode de Dubner Abate
t
10
20
30
40
50
60
70
80


ERREUR
Fonction F.5
Q Méthode de Gover Stehfest
K Méthode de Durner Abate
10
20
30
40
50
60
80

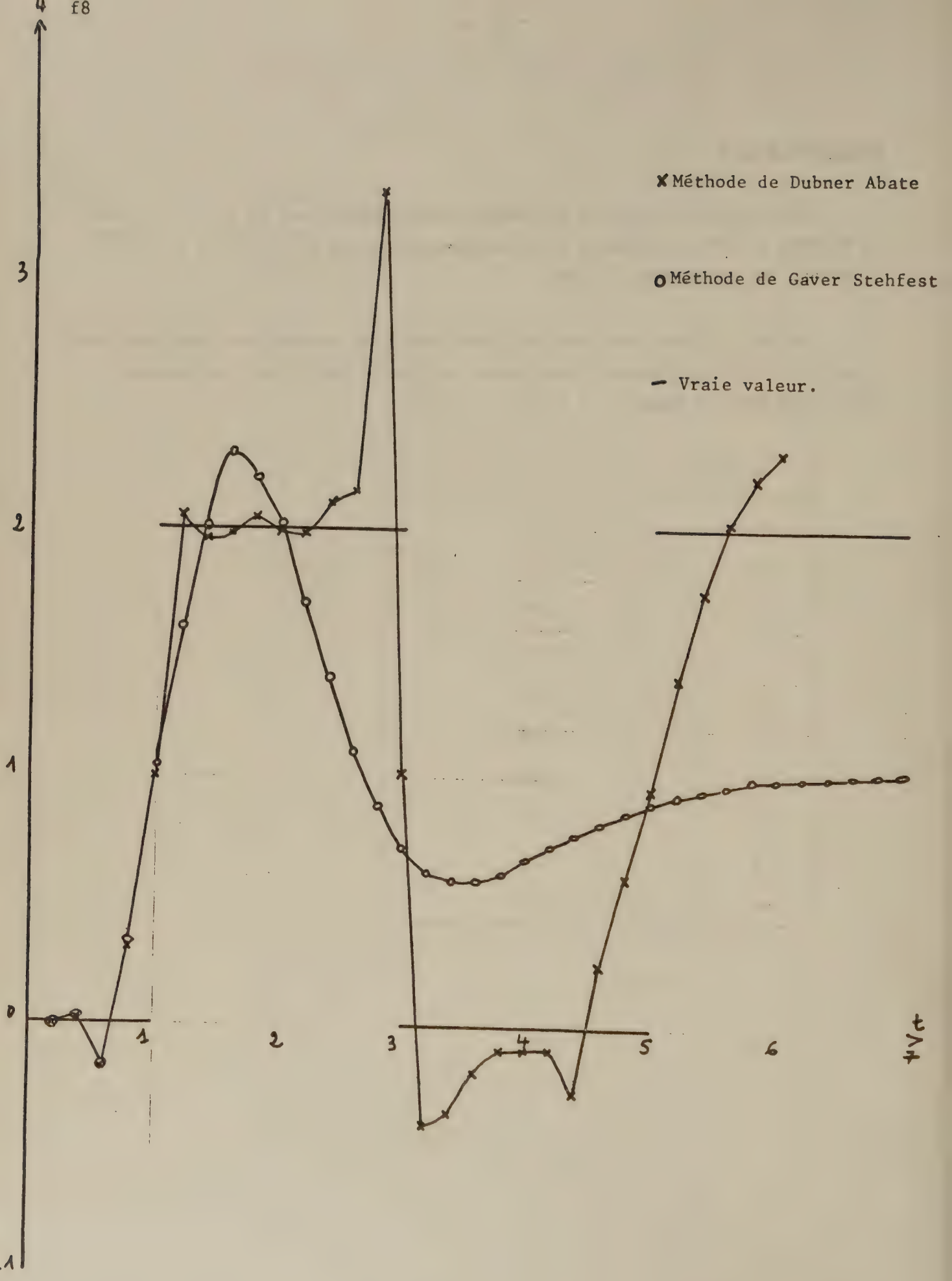

ERREUR
Fonction F16
Méthode de Gauss Stanfast
Méthode de Dutour Abate
t

Résultats pour f_7

Pour f_7 par la méthode de Dubner Abate l'erreur est de $5\ 10^{-7}$ et pour la méthode de Gaver Stehfest, elle est comprise entre $1,5\ 10^{-11}$ et $4,8\ 10^{-14}$, t variant sur un segment [1,100].

On voit l'importance des améliorations si on compare ces résultats avec ceux obtenus en programmant brutalement la formule approchée : en prenant 500 termes pour la somme et a = 0,1 on obtient :

t	f(t) = 1
1	1.04
2	1.05
3	1.05
4	1.06
5	1.07
6	1.08
7	1.09
8	1.11
9	1.13
10	1.16


f8
Méthode de Dubner Abate
Méthode de Gaver Stehfest
Vraie valeur.
4
3
2
1
0
-1
1
2
3
4
5
6
t

CHAPITRE - IX

APPLICATION A UN PROBLEME DE MECANIQUE STATISTIQUE QUANTIQUE

IX-1 - Position du problème [HR]

La mécanique quantique nous apprend que l'énergie d'un système physique (ou ensemble de molécules) ne peut varier que par quantas, c'est-à-dire prendre un ensemble discret de valeurs appelées niveaux d'énergie E_i. Cet ensemble a un élément minimum appelé énergie fondamentale E_o qui correspond à l'énergie du système à la température absolue T = 0.

On définit une densité de niveau d'énergie n, au sens des distributions de la façon suivante :

$$n = \sum_i \delta\ (E-E_i)$$

où δ est la distribution de Dirac.
A partir de là, on peut définir la distribution

$$F(E) = N\ \{E_i < E\}$$

qui a l'avantage d'être une fonction définie sur $[0,\infty[$. Elle représente le nombre de niveaux tels que l'énergie soit inférieure à E.

La mécanique statistique nous apprend que la probabilité de trouver une molécule dans l'état E_i est proportionnelle à $e^{-\beta E_i}$, où $\beta = \frac{1}{kT}$ et k la constante de Boltzmann.

On définit encore les grandeurs suivantes :

$$Z(\beta) = \langle n(E), e^{-\beta E} \rangle$$

et

$$W(\beta) = \langle F(E), e^{-\beta E} \rangle$$

qui est une transformée de Laplace.
Etant donné la relation qui existe entre n(E) et F(E) on sait que :

$$W(\beta) = \frac{Z(\beta)}{\beta}$$

Cette fonction Z(β) s'appelle fonction de partition.
.Elle est très importante pour les physiciens car sa connaissance permet de calculer l'ensemble des propriétés thermodynamiques d'un système.
Par exemple, l'énergie interne U d'un système est définie par :

$$U = \frac{1}{Z(\beta)} \int_{E_o}^{\infty} n(E)\ e^{-\beta E}\ dE$$

Or

$$-\frac{d\ Log\ Z(\beta)}{d\beta} = -\frac{Z'(\beta)}{Z(\beta)} = \frac{-\int_{E_o}^{\infty} -E\ n(E) e^{-\beta E} dE}{Z(\beta)}$$

$$= \frac{1}{Z(\beta)} \int_{E_o}^{\infty} E\ n(E)\ e^{-\beta E}\ dE$$

Il est quelque fois plus facile pour les physiciens de calculer Z(β) que n(E). D'où l'intérêt de savoir calculer l'inverse au sens de Laplace de Z(β) pour obtenir n(E).
Dans les exemples qui suivent c'est en fait $\bar{W}(E) = L^{-1}\ (\frac{Z(\beta)}{\beta})$ qui a été calculée du fait que ce sont les valeurs de cette fonction qui étaient données par Hoare et Ruijgrok dans leur article.

IV-2 - Problèmes particuliers étudiés

Nous avons étudié trois systèmes composés de N oscillateurs harmoniques ayant des énergies fondamentales :

$$\varepsilon_i = h\nu_i \qquad i = 1,2,\ldots,N$$

h constante de Planck

Dans ce cas :

$$Z(\beta) = e^{-\beta E_o} \prod_{i=1}^{N} (1-e^{-\beta\varepsilon_i})^{-1}$$

avec $E_o = \frac{1}{2} \sum_{i=1}^{N} \varepsilon_i$ = énergie fondamentale du système.

a) Le premier système est formé de molécules d'eau

$N = 3$

$\nu_1 = 3652\ cm^{-1}$

$\nu_2 = 1595\ cm^{-1}$

$\nu_3 = 3756\ cm^{-1}$

Les résultats comparés avec ceux de Hoare et Ruijgrok [16] sont les suivants :

E	$\overline{W}(E)$ exact	$\overline{W}(E)_{HR}$	$\overline{W}(E)_{DA}$
10 Xcal	3	3.54	3.2
20	11	11.23	10.2
30	23	25.52	23.8
40	46	48.43	45.8
50	78	82.00	78.2
70	183	189.0	183.7
100	466	479.5	467.4
150	1405	1444	1405.4
200	3137	3228	3137.1

$\overline{W}(E)$ ici calculée est l'intégrale de la densité de niveaux d'énergie :

$$\overline{W}(E) = L^{-1} \frac{Z(\beta)}{\beta}$$

$\overline{W}(E)_{HR}$: valeurs calculées par Hoare et Ruijgrok

$\overline{W}(E)_{DA}$: valeurs calculées par la méthode de Dubner Abate.

b) Le deuxième système est formé de molécules de deuthérométhane

$N = 9$

$\nu_1 = 2143.2 \text{ cm}^{-1}$

$\nu_2 = \nu_3 = 1109.4 \text{ cm}^{-1}$

$\nu_4 = \nu_5 = \nu_6 = 2336.8 \text{ cm}^{-1}$

$\nu_7 = \nu_8 = \nu_9 = 1026.8 \text{ cm}^{-1}$

Résultats

E	$\overline{W}(E)$ exact	$\overline{W}(E)_{HR}$	$\overline{W}(E)_{DA}$
20 Kcal	$1.19 \ 10^3$	$1.18 \ 10^3$	$1.17 \ 10^3$
50	$2.28 \ 10^5$	$2.30 \ 10^5$	$2.27 \ 10^5$
100	$3.08 \ 10^7$	$3.10 \ 10^7$	$3.07 \ 10^7$
150	$7.72 \ 10^8$	$7.18 \ 10^8$	$7.12 \ 10^8$
180	$3.08 \ 10^9$	$3.10 \ 10^9$	$3.08 \ 10^9$

c) Le troisième système est formé de molécules de cyclopropane

N = 21

ν_1 à ν_6 = 3221 cm^{-1}

ν_7 à ν_9 = 1478 cm^{-1}

ν_{10} à ν_{16} = 1118 cm^{-1}

ν_{17} à ν_{19} = 879 cm^{-1}

ν_{20} à ν_{21} = 750 cm^{-1}

Résultats

E	$\overline{W}(E)$	$\overline{W}(E)_{HR}$	$\overline{W}(E)_{DA}$
10 Kcal	$8.02\ 10^2$	$7.25\ 10^2$	$8.60\ 10^2$
20	$7.75\ 10^4$	$7.60\ 10^4$	$7.62\ 10^4$
30	$2.69\ 10^6$	$2.65\ 10^6$	$2.64\ 10^6$
40	$4.97\ 10^7$	$4.98\ 10^7$	$4.95\ 10^7$
50	$6.12\ 10^8$	$6.16\ 10^8$	$6.13\ 10^8$
100	$5.84\ 10^{12}$	$5.85\ 10^{12}$	$5.82\ 10^{12}$
150	$3.00\ 10^{15}$	$3.00\ 10^{15}$	$2.98\ 10^{15}$
200	$3.54\ 10^{17}$	$3.54\ 10^{17}$	$3.52\ 10^{17}$

Nous voyons que pour ces fonctions dont la variation est très grande les résultats sont encore bons.

CHAPITRE - X

APPLICATION A LA 'DECONVOLUTION' DE DEUX FONCTIONS

Déconvolution numérique

Nous avons vu en introduction la définition et les propriétés de la convolution de deux fonctions. Nous nous proposons de résoudre numériquement le problème suivant :

soit s(t) (s comme sortie), $T(t)$ (T comme transfert)

et e(t) (e comme entrée) trois fonctions réelles $\in T$ telles que

$$s(t) = \int_o^t T(\tau)\ e(t-\tau)d\tau$$

Nous voulons calculer $T(t)$ connaissant s(t) et e(t).

X-1 - Intérêt de ce calcul

Supposons qu'un appareil quelconque soit tel que s'il est excité par une certaine grandeur fonction du temps représenté par la fonction e(t), il transforme cette grandeur en une autre fonction du temps s(t), alors la fonction $T(t)$ telle que

$$s(t) = \int_o^t T(\tau)\ e(t-\tau)\ d\tau$$

est caractéristique de l'appareil et sa transformée de Laplace est appelée par les physiciens "fonctions de transfert" de l'appareil.

Dire que $T(t)$ ou sa transformée caractérise l'appareil signifie que tout couple $e_i(t)$, $s_i(t)$ tel que e_i soit une excitation de l'appareil et s_i une réponse, est tel que

$$s_i(t) = \int_0^t T(\tau)\ e_i(t-\tau)d\tau$$

De plus l'équation (2) s'interprête de la façon suivante :

> $T(t)$ est la réponse de l'appareil quand l'excitation qu'on lui a appliquée est la fonction de Dirac au point $t=0$. Or une telle excitation n'est pas réalisable physiquement, ce qui exclut d'obtenir $T(t)$ de cette façon. C'est pourquoi le problème qui nous intéresse ici est de calculer $T(t)$ connaissant $s(t)$ et $e(t)$ et c'est évidemment la propriété très simple qui lie les transformées de Laplace de ces trois fonctions qui va nous servir.

X-2 - Algorithme de déconvolution

Nous avons vu que notre but était de calculer $T(t)$ connaissant $s(t)$ et $e(t)$. Comme l'équation de convolution après transformation de Laplace s'écrit :

$$S(p) = T(p) \times E(p)$$

nous avons essayé de calculer numériquement $S(p)$ et $E(p)$, puis d'en faire le rapport et enfin d'essayer de calculer numériquement l'inverse de la fonction $T(p)$ ainsi obtenue.

a) Pour calculer l'inverse de $T(p)$ par la méthode de Dubner Abate nous avons vu qu'il fallait connaître les valeur de $Re[T(p)]$ pour un certain ensemble de valeurs de $p = a+ib$, a et b étant déterminés par :

- les singularités de $T(p)$ pour a
- l'intervalle sur lequel on cherche $T(t)$ pour b.

Exprimons Re[T(p)].

Posons

$$S(p) = Re[S(p)] + i\ I_m[S(p)]$$

$$E(p) = Re[E(p)] + i\ I_m[E(p)]$$

Il vient :

$$Re[S(p)]+i\ I_m[S(p)] = \{Re[T(p)]+i\ I_m[T(p)]\}\times\{Re[E(p)]+i\ I_m[E(p)]\}$$

Soit :

$$Re[T(p)]+i\ I_m[T(p)] = \frac{Re[S(p)] + i\ I_m[S(p)]}{Re[E(p)] + i\ I_m[E(p)]}$$

$$Re[T(p)] = \frac{Re[S(p)] \times Re[E(p)] + I_m[S(p)] \times I_m[E(p)]}{Re^2[E(p)] + I_m^2[E(p)]}$$

Pour calculer Re[T(p)] il suffit donc de savoir calculer les 4 termes du numérateur qui se ramènent chacun facilement à des transformées de Fourier.
En effet :

$$S(p) = \int_o^\infty e^{-pt}\ s(t)dt = \int_o^\infty e^{-(a+ib)t}\ s(t)\ dt$$

$$S(p) = \int_o^\infty e^{-at}\ e^{-ibt}\ s(t)\ dt$$

$$S(p) = \int_o^\infty e^{-at}(\cos bt - i \sin bt)\ st\ dt$$

soit :

$$Re[S(p)] = \int_o^\infty e^{-at}\ s(t)\ \cos bt\ dt$$

$$Im[S(p)] = -\int_o^\infty e^{-at}\ s(t)\ \sin bt\ dt$$

b) Pour calculer ces transformées de Fourier nous avons utilisé l'algorithme de Cooley et Tuckey que nous ne décrivons pas ici, mais que l'on trouvera aux références [CT] et [EB].
La procédure correspondante se trouve dans le programme en annexe (elle s'appelle TFCTG).

X-3 - Exemple 1

$$t = \int_0^t d\tau$$

Cet exemple ultra simple où la fonction de sortie est égale à t et la fonction d'entrée égale à 1 a permis d'essayer la méthode.

Les résultats en sont encourageants puisque pour t variant de 0,25 à 3.0 par pas de 0,25 on a obtenu $T(t) = 0,99999999$ alors que la valeur vraie est 1, soit une précision relative de 10^{-8}.

Exemple 2

$$te^{-\alpha t} = \int_0^t e^{\alpha(t-\tau)}\, e^{\alpha\tau}\, d\tau \qquad \alpha = 0.5$$

C'est pour voir la différence de comportement des calculs que nous avons chacune des fonctions de l'exemple 1 par $e^{-\alpha t}$.

La précision relative est alors comprise entre 1.4×10^{-8} et 1.3×10^{-7}.

Les résultats quoique un tout petit peu moins bons sont encore excellents.

Exemple 3

$$\frac{t^3}{6} + \frac{t^2}{2} = \int_0^t \tau(1+t-\tau)\, d\tau$$

Résultats

t	t calculé	erreur relative
0,25	0,24996	$1.71\ 10^{-4}$
0,50	0,49993	$1.33\ 10^{-4}$
0,75	0,74992	$1.04\ 10^{-4}$
1.0	0,99992	$8.06\ 10^{-5}$
1.25	1,24992	$6.27\ 10^{-5}$
1.50	1,49993	$4.87\ 10^{-5}$
1.75	1.74993	$3.79\ 10^{-5}$
2.0	1.99994	$2.94\ 10^{-5}$
2.25	2.24995	$2.29\ 10^{-5}$
2.50	2.49996	$1.77\ 10^{-5}$
2.75	2.74996	$1.38\ 10^{-5}$
3.0	2.99997	$1.07\ 10^{-5}$
4.0	3.99998	$0.4\ 10^{-5}$
5.0	4.99999	$1.6\ 10^{-6}$
6.0	5.99999	$7\ 10^{-7}$
7.0	6.99999	$3.4\ 10^{-7}$
8.0	7.99999	$2.2\ 10^{-7}$
9.0	8.99999	$1.8\ 10^{-7}$
10.0	9.99999	$1.7\ 10^{-7}$
20.0	19.99999	$0.2\ 10^{-7}$
30.0	29.99999	$0.2\ 10^{-7}$
100.0	99.99998	$0.24\ 10^{-7}$

Exemple 4

$$\frac{\cos t - \cos 3t}{8} = \int_0^t \sin(\tau) \cos 3(t-\tau)\, d\tau$$

Résultats

t	sin t vrai	sin t calculé	erreur relative
0,25	0,24740	0,24736	$1.96\ 10^{-4}$
0,50	0,47943	0,47930	$2.71\ 10^{-4}$
0,75	0,68164	0,68145	$2.82\ 10^{-4}$
1,0	0,84147	0,84103	$5.25\ 10^{-4}$
1,25	0,94898	0,94837	$6.50\ 10^{-4}$
1,50	0,99749	0,99617	$1.32\ 10^{-3}$
1,75	0,98399	0,98193	$2.09\ 10^{-3}$
2.0	0,90930	0,90624	$3.36\ 10^{-3}$
2,25	0,77807	0,77369	$5.64\ 10^{-3}$
2,50	0,59847	0,59242	$1.01\ 10^{-2}$
2,75	0,38166	0,37360	$2.11\ 10^{-2}$
3.0	0,14112	0,13075	$7.35\ 10^{-2}$

BIBLIOGRAPHIE

[BKL] R. BELLMAN, R. KALABA, J. LOCKETT
Numerical inversion of the Laplace Transform.
Elsevier - New York Londres 1966.

[BR] C. BREZINSKI
Méthodes d'accélération de la convergence en analyse numérique.
Thèse présentée à la Faculté des Sciences de Grenoble.
26 avril 1971.

[BU] BULL
Etude de la méthode de Papoulis d'inversion des transformées de Laplace et applications au problème de
Société Nationale de Recherches et d'exploitation des Pétroles en Algérie. SN REPAL - mars 1961.

[CT] COOLEY - J.W. TUKEY
An algorithm for machine calculation of complex Fourier series.
Math. Comput. 19 (1965) pp. 297-301.

[DA] H. DUBNER - J. ABATE
Numerical inversion of Laplace transform and the finite Fourier cosine transform.
Journal de l'ACM - 1968 Vol. 15 n° 1 pp. 115-123.

[EB] A. EBERHARD
Algorithmes de l'analyse harmonique numérique.
Thèse d'ingénieur docteur présentée à Grenoble le 22/6/1970

[GA] D.P. GAVER
Observing stochastic processes, and approximate transform inversion.
Operation Research 14.3 1966, pp. 444-459

[HR] R.M. HOARE - J.W. RUIJGROK
Inversion of the partition fonction : the first order steepest descent method.
The journal of chemical physics. Vol. 52 n° 1 - 1970 pp. 113-120.

[KU] FEIX - SAJALOLI - KUNTZMANN
Une variante de la méthode de Tricone Picone pour l'inversion de la transformée de Carson.
Chiffres 1958 - Vol. 1 pp. 63-74.

[MI] J. MIKUSINSKY
Operational calculus.
Pergamon Press. 1959

[NU] J.B. NUGEYRE
Inversion numérique de la transformation de Laplace-Carson
Thèse présentée à la Faculté des Sciences de Grenoble
13 juin 1960.

[PA] A. PAPOULIS
A new method of inversion of the Laplace transform.
Quaterly of applied mathematics. Vol. 14 1956 pp. 405-414.

[RI] G. RIBIERE
Amélioration du résidu dans la résolution des systèmes linéaires au sens des moindres carrés.
Thèse de 3ème Cycle Paris 1966.

[ST] H. STEHFEST
Numerical inversion of Laplace transforms.
Communication de l'ACM
Vol. 13 n° 1 - 1970 - pp. 47-49.

[TI] TICHONOFF
Dokl Akad. Nauk 151 (1963) pp. 501-504
153 (1963) pp. 49-52.

[WY1] P. WYNN
On a procrustean technique for the numerical transformation of slowly convergent sequences and series.
Proc. Camb. Phil. Soc. Vol. 52 pp. 663-671 - 1956.

[WY2] P. WYNN
On the convergence and stability of the ε-algorithm.
SIAM Num. Anal. Vol. 3 pp. 91-122.

[YO] K. YOSIDA
Fonctional analysis.
Die grundelehren der mathematischen wissenshafsten in einzeldarstellungen. Baud 123.
Springer Verlag Berlin-Gottingen-Heidelberg 1965.

[ZA] M. ZAMANSKY
Introduction à l'algèbre et l'analyse modernes.
Dunod 1958.

[ZBF] BALBINE et FRANKLIN
The calculation of Fourier Integrals.
Math. Comp. 20 (1966) pp. 570-589.

TABLE DES MATIERES
-:-:-:-:-:-:-:-:-:

INTRODUCTION

CONVOLUTION - PROPRIETES - OPERATEURS DE MIKUSINSKY

I - Convolution de fonctions de l'espace vectoriel $\mathcal{C}$: fonctions à valeurs réelles, définies et continues sur $[0,\infty[$ --------- 1

II - $\mathcal{C}$ est un anneau intègre par rapport aux opérations d'addition et de convolution ---------- 2

III - Construction d'un corps des quotients et conventions de notation 3

IV - L'opérateur d'intégration h ---------- 5

V - L'opérateur scalaire ---------- 5

VI - L'opérateur de différenciation ---------- 6

VII - Liaison avec la transformée de Laplace ---------- 7

VIII - Fonctions de classe $\mathcal{K}$ ---------- 8

IX - Egalité de deux fonctions de classe $\mathcal{K}$ considérées comme opérateurs ---------- 9

X - Puissances non entières de l'opérateur h ---------- 10

XI - Fonctions ayant une dérivée de classe $\mathcal{K}$ ---------- 10

XII - Echelon et opérateur de translation k^{λ} ---------- 11

XIII - Fonctions à valeurs opérateurs et leurs dérivées ---------- 13

XIV - Décomposition d'une fonction à valeurs opérateurs ---------- 14

XV - Décomposition sur un intervalle infini ---------- 15

XVI - Dérivée décomposable d'une fonction à valeurs opérateurs --- 16

XVII - Propriétés de la dérivée décomposable des fonctions à valeurs opérateurs ---------- 17

XVIII - Dérivées décomposables d'ordre supérieur ---------- 18

IXX - Dérivée décomposable sur un intervalle infini ---------- 18

XX - Fonction exponentielle généralisée ---------- 19

XXI - Théorème sur l'expression de la transformée de Laplace au moyen des opérateurs ---------- 20

XXII - Suites de fonctions - Convergence uniforme et presque uniforme ---------- 21

XXIII - Définition de la convergence de suites d'opérateurs : convergence ---------- 22

XXIV - Continuité de la transformation de Laplace ---------- 23

XXV - Continuité de la transformation inverse ---------- 23

XXVI - Difficultés du calcul numérique de la transformation inverse 24

CHAPITRE - I

METHODES NUMERIQUES D'INVERSION DE LA TRANSFORMEE DE LAPLACE

METHODE DE PAPOULIS UTILISANT LES POLYNOMES DE LEGENDRE

I-1 Rappel de la méthode ---------- 26

I-2 Modalités d'emploi de la méthode ---------- 29

CHAPITRE - II

METHODE DE PAPOULIS UTILISANT LES POLYNOMES DE LAGUERRE

II-1 Rappel de la méthode ---------- 30

II-2 Exemple comparé avec les résultats d'autres méthodes --------- 32

CHAPITRE - III

METHODE DE REGULARISATION

III-1 Discrétisation de l'opérateur d'intégration ---------- 35

III-2 Ecriture du système linéaire ---------- 37

III-3 Résolution du système linéaire ---------- 37

III-4 Conclusions ---------- 39

III-5 Raffinement de la méthode ---------- 40

III-6 Exemple comparé avec les résultats d'autres méthodes --------- 41

CHAPITRE - IV

METHODE DE BELLMAN : PROGRAMMATION DYNAMIQUE

IV-1 Exposé de la méthode ---------- 43

IV-2 Cas où une approximation est déjà connue ---------- 44

IV-3 Première approche de la programmation dynamique ---------- 45

IV-4 Détermination des valeurs minimisantes ---------- 47

IV-5 Approximation successives ---------- 48

IV-6 Obtention d'une approximation initiale ---------- 49

IV-7 Développement de l'approximation initiale en série de puissances 49

IV-8 Approximations successives avec lissage ---------- 50

IV-10 Exemple comparé avec les résultats des autres méthodes -------- 55

CHAPITRE - V

METHODE DE GAVER STEHFEST

V-1 Exposé de la méthode ---------- 56

V-2 Procédure et exemple d'activation ---------- 61

V-3 Considérations pratiques sur cette méthode ---------- 64

CHAPITRE - VI

METHODE DE DUBNER ABATE

VI-1 Exposé de la méthode ---------- 66

VI-2 Conclusions ---------- 69

VI-3 Approfondissement sur l'erreur ---------- 69

CHAPITRE - VII

ETUDE DETAILLEE ET AMELIORATIONS APPORTEES A LA METHODE DE DUBNER ABATE

VII-1 Etude de l'erreur ---------- 71

VII-2 Méthode d'accélération de la convergence de la somme ε- algorithme 72

VII-3 Etude du cas simple f(t) = 1 ---------- 76

a) Relation entre les différents paramètres ---------- 77

b) Influence du nombre N auquel on arrête la sommation -------- 78

c) Blocage des termes de la somme ---------- 80

d) Valeur optimale du rapport de certains paramètres ---------- 81

VII-4 Recherche de l'optimum d'un des paramètres ---------- 82

a) Algorithme ---------- 82

b) Résultats ---------- 83

c) Organigramme ---------- 84

CHAPITRE - VIII

RESULTATS COMPARES DES DIFFERENTES METHODES ---------- 88

CHAPITRE - IX

APPLICATION A UN PROBLEME DE MECANIQUE STATISTIQUE QUANTIQUE

IX-1 Position du problème ---------- 89

IX-2 Cas particuliers étudiés - Résultats comparés ---------- 91

CHAPITRE - X

APPLICATION A LA DECONVOLUTION DE DEUX FONCTIONS

X-1 Intérêt de la déconvolution ---------- 94

X-2 Algorithme de déconvolution ---------- 95

X-3 Exemples et résultats ---------- 97

BIBLIOGRAPHIE ---------- 100

ANNEXE - I

-:-:-:-:-:

PROGRAMME D'INVERSION

```
FILE: XYZ      MEMO      P1                    CAMBRIDGE MONITOR SYSTEM

//TISAG JOB (6053,0154),'F.VEILLON',MSGLEVEL=1,CLASS=C
// EXEC ALGOCLG,PARM.ALGOL='L2'
//ALGOL.SYSIN DD *
'BEGIN' 'PROCEDURE' LAPLACETINVERSE(REIG,TE,NTE,AE,NAE,ITERE,RESULTATE) ;
'REAL' 'PROCEDURE' REIE ; 'REAL' 'ARRAY' TE,AE,RESULTATE ;
'INTEGER' ITERE,NTE,NAE ;
'COMMENT' CETTE PROCEDURE CALCULE ET RANGE DANS LE TABLEAU RESULTATE LES
 NTE VALEURS DE LA FONCTION F(T) POUR LES ABSCISSES CONTENUES DANS LE
TABLEAU TE. LES NAE VALEURS DU PARAMETRE A SONT DANS LE TABLEAU AE.
ITERE EST LE NOMBRE DE TERMES PRIS EN CONSIDERATION DANS LA
SOMME INFINIE DIVISE PAR 8 ;
'BEGIN' 'PROCEDURE' EPSALGOR(EPS,NEPS,RESULEPS) ;
    'ARRAY' EPS,RESULEPS ; 'INTEGER' NEPS ;
    'COMMENT' CETTE PROCEDURE CALCULE DES VALEURS APPROCHEES DE LA SOMME
DE LA SUITE INFINIE A PARTIR DES NEPS PREMIERS TERMES DE LA
SERIE ASSOCIEE QUI SONT RANGES DANS EPS. LES VALEURS SONT RANGEES
DANS RESULEPS. C'EST LA VALEUR DE L'ELEMENT DE RESULEPS DE
PLUS GRAND INDICE QUI EST LA SOMME APPROCHEE CHERCHEE ;
    'BEGIN' 'ARRAY' EPSTAMP(/1:NEPS-1/) ; 'INTEGER' I,J,K ;
        'FOR' I:=1 'STEP' 1 'UNTIL' NEPS '/' 2 'DO'
          RESULEPS(/I/):=0.0 ;
        'FOR' I:=1 'STEP' 1 'UNTIL' NEPS-1 'DO'
        'BEGIN' 'IF' EPS(/I+1/)=EPS(/I/) 'THEN'
        OUTSTRING(1,'('TERMES EGAUX')') ;
          EPSTAMP(/I/):=1.0/(EPS(/I+1/)-EPS(/I/)) ;
        'END' ;
        RESULEPS(/1/):=EPS(/NEPS/) ;
        K:=2 ;
        'FOR' J:=NEPS-2 'STEP' -1 'UNTIL' 1 'DO'
        'BEGIN' 'FOR' I:=1 'STEP' 1 'UNTIL' J 'DO'
          'BEGIN' EPS(/I/):=EPSTAMP(/I/) ;
            'IF' EPSTAMP(/I+1/)=EPSTAMP(/I/) 'THEN' 'GOTO' FIN ;
            EPSTAMP(/I/):=EPS(/I+1/)+1.0/(EPSTAMP(/I+1/)-EPSTAMP(/I/)) ;
          'END' ;
          'IF' (K '/' 2)*2=K 'THEN' RESULEPS(/(K'/' 2)+1/):=EPSTAMP(/J/);
          K:=K+1 ;
    'END' ;
    FIN: 'END' EPSALGOR ;
    'PROCEDURE' LAPLINV(REIG,TG,ITERG,AG,RESULTATG) ;
    'REAL' 'PROCEDURE' REIG ; 'REAL' TG,RESULTATG,AG ;
    'INTEGER' ITERG ;
    'COMMENT' CETTE PROCEDURE CALCULE DANS RESULTATG LA VALEUR DE F(T),
    T ETANT DANS TG,ET LE PARAMETRE A ETANT DANS AG. ITERG A LE
    MEME SENS QUE ITERE;
    'BEGIN' 'REAL' SOMME ; 'INTEGER' I,J ;
        'REAL' 'ARRAY' ETAB(/0:8*ITERG/),FD(/1:ITERG/),
      RESULEP(/1:ITERG '/' 2/) ;
        'FOR' I:=0 'STEP' 1 'UNTIL' 8*ITERG 'DO'
          ETAB(/I/):=REIG(AG,I*3.1415926536/(8.0*TG)) ;
        SOMME:=0.0 ;
        'FOR' I:=1 'STEP' 1 'UNTIL' ITERG 'DO'
        'BEGIN' 'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
          SOMME:=SOMME+ETAB(/J+8*(I-1)/)*COS((J+8*(I-1))*
          3.1415926536/8.0) ;
          FD(/I/):=SOMME ;
```

```
      'END' ;
      RESALCOR(ER,STERC,RESULER) ;
      RESULTATC:=2.0*EXP(AG*TG)/(P.0*TG)*(RESULER(/ITERC '/' 2/)
         +0.5*TTAB(/0/)) ;
   'END' LABLINV ;
   'INTEGER' I,J ;
   'REAL' A1,B1,C1,ZERC,X1,X2 ;
   'REAL' 'ARRAY' X,Y,M,T,U,Z(/0:MAF-1/) ;
'PROCEDURE' COEFSPLINETROIS(N,X,Y,M);
'VALUE' N,X,Y; 'INTEGER' N; 'ARRAY'X,Y,M;
'BEGIN' 'INTEGER' I; 'ARRAY'D(/1: N-1/) ; 'REAL' A,B,C,E ;
      'FOR' I:=1 'STEP' 1 'UNTIL' N-1 'DO'
    'BEGIN' A:=X(/I+1/)-X(/I/);
    B:=X(/I/)-X(/I-1/); C:=Y(/I+1/)-Y(/I/);
       E:=Y(/I/)-Y(/I-1/); 'IF' I=N-1 'THEN'
         'BEGIN' D(/I/):=(X(/I+1/)-X(/I-1/))/3;
       M(/I/):=C/A-E/B
       'END' 'ELSE' 'BEGIN' D(/I/):=(12*D(/I+1/)*(X(/I+1/)-X(/I-1/))
        -A*B)/(36*D(/I+1/));
          M(/I/):=C/A-E/B-A*M(/I+1/)/(6*D(/I+1/))
          'END'                                                    0205
   'END' ; M(/0/):=M(/N/):=.0 ;
   'FOR' I:=1 'STEP' 1 'UNTIL' N-1 'DO'
   'IF' I=1 'THEN' M(/I/):=M(/I/)/D(/I/) 'ELSE'
      M(/I/):=(6*M(/I/)-(X(/I/)-X(/I-1/))*M(/I-1/))/(6*D(/I/))
    'END' COEFSPLINETROIS ;
  'FOR' I:=1 'STEP' 1 'UNTIL' NTE 'DO'
  'BEGIN' 'FOR' J:=0 'STEP' 1 'UNTIL' MAF-1 'DO'
    'BEGIN' X(/J/):=AF(/I/)/TE(/I/) ;
      LABLINV(PELE,TE(/I/),TTERE,X(/J/),Y(/J/)) ;
      RESULTATE(/I/):=Y(/J/) ;
    'END' ;
    'IF' MAF >= 1 'THEN'
    'BEGIN' COEFSPLINETROIS(MAF-1,X,Y,M) ;
      'FOR' J:=0 'STEP' 1 'UNTIL' MAF-2 'DO'
      'BEGIN' A1:=(M(/J+1/)-M(/J/))/6.0/(X(/J+1/)-X(/J/)) ;
          B1:=(M(/J/)-6.0*A1*X(/J/))/2.0 ;
          C1:=(Y(/J+1/)-Y(/J/))/(X(/J+1/)-X(/J/))-A1
             (X(/J/)*X(/J/)+X(/J+1/)*X(/J+1/)+X(/J/)*X(/J+1/))
             -B1*(X(/J/)+X(/J+1/)) ;
        'IF' B1*B1-3.0*A1*C1 >= 0.0 'THEN'
        'BEGIN' 'IF' A1=0.0 'AND' B1 >= 0.0 'THEN'
          'BEGIN' X2:=-C1/2.0/B1 ;
            'GOTO' TRUC ;
          'END' ;
          'IF' A1=0.0 'AND' B1=0.0 'THEN' 'GOTO' E3 ;
          X1:=(-B1+SQRT(B1*B1-3.0*A1*C1))/A1/3.0 ;
          X2:=(-B1-SQRT(B1*B1-3.0*A1*C1))/A1/3.0 ;
          'IF' X(/J/) < X1 'AND' X1 < X(/J+1/) 'THEN'
          'BEGIN' U(/MAF-2/):=X1 ;
            'GOTO' E1 ;
          'END' ;
          TRUC: 'IF' X(/J/) < X2 'AND' X2 < X(/J+1/) 'THEN'
          'BEGIN' U(/MAF-2/):=X2 ;
            'GOTO' E1 ;
```

```
FILE: XYZ      MEMO      P1                    CAMBRIDGE MONITOR SYSTEM

                'END' ;
              'END';
               E3: Z(/J/):=ABS((3.0*A1*X(/J/)+2.0*B1)*X(/J/)+C1);
               T(/J+1/):=ABS((3.0*A1*X(/J+1/)+2.0*B1)*X(/J+1/)+C1);
               'IF' ABS(Z(/J/)) < ABS(Z(/J+1/)) 'THEN'
               'BEGIN' T(/J/):=ABS(Z(/J/)) ;
                  U(/J/):=X(/J/) ;
               'END' 'ELSE'
               'BEGIN' T(/J/):=ABS(Z(/J+1/)) ;
                  U(/J/):=X(/J+1/) ;
               'END' ;
               'IF' A1=0.0 'AND' B1=0.0 'THEN' 'GOTO' E2 ;
               ZERO:=-B1/3.0/A1 ;
               'IF' ZERO > X(/J/) 'AND' ZERO < X(/J+1/) 'THEN'
               'BEGIN' T(/J/):=ABS((3.0*A1*ZERO+2.0*B1)*ZERO+C1);
                  U(/J/):=ZERO ;
               'END' ;
            E2: 'END' ;
            'FOR' J:=0 'STEP' 1 'UNTIL' NAF-3 'DO'
               'IF' T(/J+1/) > T(/J/) 'THEN'
               'BEGIN' U(/J+1/):=U(/J/) ;
                  T(/J+1/):=T(/J/) ;
               'END' ;
         'END' ;
         E1:LAPLINV(EPSF,TF(/I/),ITERF,U(/NAF-2/),Y(/NAF-2/)) ;
         RESULTATF(/I/):=Y(/NAF-2/) ;
      'END' ;
   'END' LAPLACEINVERSE ;
   'REAL''PROCEDURE'F1(A,B);'REAL'A,B;
   'BEGIN'F1:=EXP(-25.0*A)*(A*COS(25.0*B)
   -B*SIN(25.0*B))/(A*A+B*B);
   'END';
   'INTEGER' NA,NT ;
   ININTEGER(0,NA) ; 'COMMENT' LECTURE DU NOMBRE DES AI ;
   ININTEGER(0,NT) ; 'COMMENT' LECTURE DU NOMBRE DES VALEURS DE T ;
   'BEGIN' 'INTEGER' I,ITER ;
      'REAL' 'ARRAY' A(/0:NA-1/),T,RESULTAT(/1:NT/);
      ININTEGER(0,ITER) ;
      OUTSTRING(1,'(' ITER=')');
      OUTINTEGER(1,ITER);
      SYSACT(1,14,1);
      OUTSTRING(1,'(' ')');
      'FOR' I:=0 'STEP' 1 'UNTIL' NA-1 'DO'
      'BEGIN' INREAL(0,A(/I/));
         OUTSTRING(1,'('VALEUR DES AI')');
         OUTREAL(1,A(/I/));
         SYSACT(1,14,1);
         OUTSTRING(1,'(' ')');
      'END';
         'COMMENT' LECTURE DES AI ;
      'FOR' I:=1 'STEP' 1 'UNTIL' NT 'DO'
         INREAL(0,T(/I/)) ;
      'COMMENT' LECTURE DES VALEURS DE T ;
      LAPLACEINVERSE(0.1,T,NT,A,NA,ITER,RESULTAT) ;
      'FOR' I:=1 'STEP' 1 'UNTIL' NT 'DO'
```

```
FILE: XYZ     MEND     P1                          CAMBRIDGE MONITOR  SYSTEM

   'BEGIN' 'COMMENT' TABULATION DES VALEURS DE T ET DE F(T) ;
      OUTSTRING(1,'('T=')') ;
      OUTREAL(1,T(/T/)) ;
      OUTSTRING(1,'('F(T)=')') ;
      OUTREAL(1,RESULTAT(/T/)) ;
      SYSACT(1,14,2) ;
OUTSTRING(1,'('  ')');
   'END' ;
'END' ;
'END' ;
```

ANNEXE - II

-:-:-:-:-:-

PROGRAMME DE DECONVOLUTION

```
SC   SOURCE STATEMENT
C0000   'BEGIN' 'PROCEDURE' LAPLACEDLCONVCL(ENTREEF,SORTIEF,TF,NTF,AF,NAF,
C0000                ITERF,RESULTATF,PF,QF);
00001   'REAL' 'PROCEDURE' ENTREEF,SORTIEF;
C0002   'REAL' 'ARRAY' TF,AF,RESULTATF;
C0003   'INTEGER' ITERF,NTF,NAF,PF,QF;
00004   'COMMENT' CETTE PROCEDURE CALCULE ET RANGE DANS LE TABLEAU RESULTATF LES
00004    NTF VALEURS DE LA FONCTION F(T) POUR LES ABSCISSES CONTENUES DANS LE
C0004   TABLEAU TF. LES NAF VALEURS DU PARAMETRE A SONT DANS LE TABLEAU AF.
00004   ITERF EST LE NOMBRE DE TERMES PRIS EN CONSIDERATION DANS LA
C0004   SOMME INFINIE DIVISE PAR 8 ;
C0004   'BEGIN' 'PROCEDURE' EPSALGOR(EPS,NEPS,RESULEPS) ;
C0005      'ARRAY' EPS,RESULEPS ; 'INTEGER' NEPS ;
C0007      'COMMENT' CETTE PROCEDURE CALCULE DES VALEURS APPROCHEES DE LA SOMME
C0007   DE LA SUITE INFINIE A PARTIR DES NEPS PREMIERS TERMES DE LA
C0007   SERIE ASSOCIEE QUI SONT RANGES DANS EPS. LES VALEURS SONT RANGEES
C0007   DANS RESULEPS. C'EST LA VALEUR DE L'ELEMENT DE RESULEPS DE
C0007   PLUS GRAND INDICE QUI EST LA SOMME APPROCHEE CHERCHEE ;
00007      'BEGIN' 'ARRAY' EPSTAMP(/1:NEPS-1/) ; 'INTEGER' I,J,K ;
C0009         'FOR' I:=1 'STEP' 1 'UNTIL' NEPS '/' 2 'DO'
0C009            RESULEPS(/I/):=0.0 ;
C0010         'FOR' I:=1 'STEP' 1 'UNTIL' NEPS-1 'DO'
C0010         'BEGIN' 'IF' EPS(/I+1/)=EPS(/I/) 'THEN'
C0010            OUTSTRING(1,'('TERMES EGAUX')') ;
00011            EPSTAMP(/I/):=1.0/(EPS(/I+1/)-EPS(/I/)) ;
C0012         'END' ;
C0013         RESULEPS(/1/):=EPS(/NEPS/) ;
C0014         K:=2 ;
00015         'FOR' J:=NEPS-2 'STEP' -1 'UNTIL' 1 'DO'
00015         'BEGIN' 'FOR' I:=1 'STEP' 1 'UNTIL' J 'DO'
00015            'BEGIN' EPS(/I/):=EPSTAMP(/I/) ;
00016               'IF' EPSTAMP(/I+1/)=EPSTAMP(/I/) 'THEN' 'GOTO' FIN ;
00017               EPSTAMP(/I/):=EPS(/I+1/)+1.0/(EPSTAMP(/I+1/)-EPSTAMP(/I/)) ;
C0018            'END' ;
00019            'IF' (K '/' 2)*2=K 'THEN' RESULEPS(/(K'/' 2)+1/):=EPSTAMP(/J/);
C0020            K:=K+1 ;
00021         'END' ;
C0022      FIN: 'END' EPSALGOR ;
C0023   'PROCEDURE' TECTC(V,N,W,M,TT,SIGNE) ;
00024   'INTEGER' N,M ;
00025   'REAL' TT,SIGNE ;
C0026   'ARRAY' V,W ;
00027   'BEGIN'
00027      'INTEGER' PN2,PM2,PNM2,PMN2,H,I,J,K,L,ISL,PK2,PMK2,PKM2,JPK2,J1,
C0027                                                KPNM2,J2,IPMK2,S ;
00028      'REAL' DEPI,X,CO,SI,COI,SII,CO1I,SI1I,CO11J,SI1J,SIIJ,C1,C2,V1HJ1,
C0026         V10,V20,V1K,V2K,TSPN2,COIJ,V1HJ2,V2HJ1,V2HJ2,VRS,VRI,WW1,WW2,
```

```
 SC      SOURCE STATEMENT

00028                                                    V11K,V21K,V1K1,V2K1 ;
00029          PN2 := 2**N ;
00030          TSPN2 := TT/PN2 ;
00031          PM2 := 2**M ;
00032          PNMM2 := PN2'/'PM2 ;
00033          PMM12 := PM2'/'2 ;
00034          DEPI := 6.283185 ;
00035          'BEGIN'
00035               'INTEGER' 'ARRAY' IND(/0:PM2-1/) ;
00036               'ARRAY' VR(/1:2,0:PNMM2-1,0:PM2-1/),T(/1:2,0:PMM12-1/) ;
00037
00037     'COMMENT' V(K) = 2*V(K) - V(K+1) - V(K-1)
00037               ET REORDONNANCEMENT DE V DANS VR ;
00037
00037               V10 := V11K := V1K := V(/1,0/) ;
00038               V20 := V21K := V2K := V(/2,0/) ;
00039               WW1 := - V10/2.0 ;
00040               WW2 := - V20/2.0 ;
00041               'FOR' K := 0 'STEP' 1 'UNTIL' PN2 - 1 'DO'
00041               'BEGIN'
00041                    WW1 := WW1 + V1K ;
00042                    WW2 := WW2 + V2K ;
00043                    V1K1 := V(/1,K+1/) ;
00044                    V2K1 := V(/2,K+1/) ;
00045                    V(/1,K/) := 2.0*V1K - V11K - V1K1 ;
00046                    V(/2,K/) := 2.0*V2K - V21K - V2K1 ;
00047                    V11K := V1K ;
00048                    V21K := V2K ;
00049                    V1K := V1K1 ;
00050                    V2K := V2K1
00050               'END' ;
00051               W(/1,0/) := (WW1 + V1K1/2.0)*TSPN2 ;
00052               W(/2,0/) := (WW2 + V2K1/2.0)*TSPN2 ;
00053               IND(/0/) := J := 0 ,
00054               VR(/1,0,0/) := V(/1,0/) ;
00055               VR(/2,0,0/) := V(/2,0/) ;
00056               S := PM2 ,
00057               'FOR' S := S'/'2 'WHILE' S >= 1 'DO'
00057               'BEGIN'
00057                    ISL := PMM12'/'S ,
00058                    'FOR' L := 1 'STEP' 1 'UNTIL' ISL 'DO'
00058                    'BEGIN'
00058                         J := J + 1 ,
00059                         IND(/J/) := K := IND(/L-1/) + S ,
00060                         KPNMM2 := K*PNMM2 ;
00061                         VR(/1,0,J/) := V(/1,KPNMM2/) ;
00062                         VR(/2,0,J/) := V(/2,KPNMM2/)
00062                    'END'
```

```
SC    SOURCE STATEMENT
0062          'END' ;
0063          'FOR' J := 0 'STEP' 1 'UNTIL' PM2-1 'DO'
0063          'BEGIN'
0063               L := IND(/J/)*PMM2 ;
0064               'FOR' I := 1 'STEP' 1 'UNTIL' PMM2-1 'DO'
0064               'BEGIN'
0064                    VR(/1,I,J/) := V(/1,L+I/) ;
0065                    VR(/2,I,J/) := V(/2,L+I/)
0065               'END'
0065          'END' ;
0066
0066    'COMMENT' INITIALISATION DE T ;
0066
0066          T(/1,0/) := CO1I := 1.0 ;
0067          T(/2,0/) := SI1I := 0.0 ;
0068          X := DEPI/PM2*SIGNE ;
0069          CO := COS(X) ;
0070          SI := SIN(X) ;
0071          'FOR' I := 1 'STEP' 1 'UNTIL' PM12 - 1 'DO'
0071          'BEGIN'
0071               T(/2,I/) := SII := SI1I*CO + CO1I*SI ;
0072               T(/1,I/) := CO1I := CO1I*CO - SI1I*SI ;
0073               SI1I := SII
0073          'END' I ;
0074
0074    'COMMENT' ALGORITHME ;
0074
0074          PK2 := 1 ;
0075          PMMK2 := PM2 ;
0076          'FOR' K := 1 'STEP' 1 'UNTIL' M 'DO'
0076          'BEGIN'
0076               PKM12 := PK2 ;
0077               PK2 := PK2*2 ;
0078               PMMK2 := PMMK2'/'2 ;
0079               IPMMK2 := 0 ;
0080               'FOR' I := 0 'STEP' 1 'UNTIL' PKM12 - 1 'DO'
0080               'BEGIN'
0080                    C1 := T(/1,IPMMK2/) ;
0081                    C2 := T(/2,IPMMK2/) ;
0082                    JPK2 := 0 ;
0083                    'FOR' J := 0 'STEP' 1 'UNTIL' PMMK2 - 1 'DO'
0083                    'BEGIN'
0083                         J1 := JPK2 + I ;
0084                         J2 := J1 + PKM12 ;
0085                         'FOR' H := 0 'STEP' 1 'UNTIL' PMM2 - 1 'DO'
0085                         'BEGIN'
0085                              V1HJ1 := VR(/1,H,J1/) ;
0086                              V1HJ2 := VR(/1,H,J2/) ;
```

```
SC      SOURCE STATEMENT
CC087                                   V2HJ1 := VR(/2,H,J1/) ;
00088                                   V2HJ2 := VR(/2,H,J2/) ;
00089                                   VRR := C1*V1HJ2 - C2*V2HJ2 ;
CC090                                   VRI := C2*V1HJ2 + C1*V2HJ2 ;
00091                                   VR(/1,H,J1/) := V1HJ1 + VRR ;
CC092                                   VR(/2,H,J1/) := V2HJ1 + VRI ;
CC093                                   VR(/1,H,J2/) := V1HJ1 - VRR ;
CC094                                   VR(/2,H,J2/) := V2HJ1 - VRI
CC094                               'END' H ;
CC095                               JPK2 := JPK2 + PK2
CC095                           'END' J ;
CC096                           IPMMK2 := IPMMK2 + PMMK2
CC096                       'END' I
CC096                   'END' K ;
CC097                   X := DEPI/PN2*SIGNE ;
CC098                   CO := COII := COS(X) ;
CC099                   SI := SIII := SIN(X) ;
CC100                   'FOR' I := 1 'STEP' 1 'UNTIL' PM2 - 1 'DO'
CC100                   'BEGIN'
00100                       COI := COII ;
CC101                       SII := SIII ;
00102                       COI1J := 1.0 ;
00103                       SII1J := WW1 := WW2 := 0.0 ;
00104                       'FOR' J := 0 'STEP' 1 'UNTIL' PMM2 - 1 'DO'
CC104                       'BEGIN'
CC104                           COIJ := COI1J ;
CC105                           SIIJ := SII1J ;
00106                           VRR := VR(/1,J,I/) ;
CC107                           VRI := VR(/2,J,I/) ;
CC108                           WW1 := WW1 + COIJ*VRR - SIIJ*VRI ;
00109                           WW2 := WW2 + SIIJ*VRR + COIJ*VRI ;
CC110                           COI1J := COIJ*COI - SIIJ*SII ;
CC111                           SII1J := COIJ*SII + SIIJ*COI
CC111                       'END' J ;
00112                       COII := COI*CO - SII*SI ;
CC113                       SIII := COI*SI + SII*CO ;
00114                       C1 := TT*PN2/(DEPI*I)**2 ;
00115                       C2 := TT/(DEPI*I)*SIGNE ;
00116                       W(/1,I/) := C1*(V1K - V11K + WW1) + C2*(V2K - V20) ;
00117                       W(/2,I/) := C1*(V2K - V21K + WW2) - C2*(V1K - V10)
CC117               'END' I
00117           'END'
00117       'END' TFCTG ;
00118       'PROCEDURE' LAPLINV(RFIG,TG,ITERG,AG,RESULTATG) ;
00119                   'ARRAY' RFIG ; 'REAL' TG,RESULTATG,AG ;
00121       'INTEGER' ITERG ;
00122       'COMMENT' CETTE PROCEDURE CALCULE DANS RESULTATG LA VALEUR DE F(T),
00122       T ETANT DANS TG,ET LE PARAMETRE A ETANT DANS AG. ITERG A LE
```

```
SC      SOURCE STATEMENT
00160             TFCTG(F,PF,FR,GF,16.0*TF(/I/),-1.0) ;
00161             TFCTG(H,PF,HR,GF,16.0*TF(/I/),-1.0) ;
00162             'FOR' K2:=0 'STEP' 1 'UNTIL' K3 'DO'
00162                RF1F(/K2/):=(FR(/1,K2/)*HR(/1,K2/)+FR(/2,K2/)*
00162                  HR(/2,K2/))/(HR(/1,K2/)*HR(/1,K2/)+HR(/2,K2/)*
00162                  HR(/2,K2/));
00163             LAPLINV(RF1F,TF(/I/),ITERF,X(/J/),Y(/J/)) ;
00164             RESULTATF(/I/):=Y(/J/) ;
00165          'END' ;
00166          'IF' NAF ¬= 1 'THEN'
00166          'BEGIN' COEFSPLINETSOIS(NAF-1,X,Y,M) ;
00167             'FOR' J:=0 'STEP' 1 'UNTIL' NAF-2 'DO'
00167             'BEGIN' A1:=(M(/J+1/)-M(/J/))/6.0/(X(/J+1/)-X(/J/)) ;
00168                   B1:=(M(/J/)-6.0*A1*X(/J/))/2.0 ;
00169                   C1:=(Y(/J+1/)-Y(/J/))/(X(/J+1/)-X(/J/))-A1*
00169                     (X(/J/)*X(/J/)+X(/J+1/)*X(/J+1/)+X(/J/)*X(/J+1/))
00169                     -B1*(X(/J/)+X(/J+1/)) ;
00170                'IF' B1*B1-3.0*A1*C1 >= 0.0 'THEN'
00170                'BEGIN' 'IF' A1=0.0 'AND' B1 ¬= 0.0 'THEN'
00170                   'BEGIN' X2:=-C1/2.0/B1 ;
00171                      'GOTO' TRUC ;
00172                   'END' ;
00173                   'IF' A1=0.0 'AND' B1=0.0 'THEN' 'GOTO' E3 ;
00174                   X1:=(-B1+SQRT(B1*B1-3.0*A1*C1))/A1/3.0 ;
00175                   X2:=(-B1-SQRT(B1*B1-3.0*A1*C1))/A1/3.0 ;
00176                   'IF' X(/J/) < X1 'AND' X1 < X(/J+1/) 'THEN'
00176                   'BEGIN' U(/NAF-2/):=X1 ;
00177                      'GOTO' E1 ;
00178                   'END' ;
00179                   TRUC: 'IF' X(/J/) < X2 'AND' X2 < X(/J+1/) 'THEN'
00179                   'BEGIN' U(/NAF-2/):=X2 ;
00180                      'GOTO' E1 ;
00181                   'END' ;
00182             'END';
00183             E3: Z(/J/):=ABS((3.0*A1*X(/J/)+2.0*B1)*X(/J/)+C1);
00184             Z(/J+1/):=ABS((3.0*A1*X(/J+1/)+2.0*B1)*X(/J+1/)+C1);
00185             'IF' ABS(Z(/J/)) < ABS(Z(/J+1/)) 'THEN'
00185             'BEGIN' T(/J/):=ABS(Z(/J/)) ;
00186                U(/J/):=X(/J/) ;
00187             'END' 'ELSE'
00187             'BEGIN' T(/J/):=ABS(Z(/J+1/)) ;
00188                U(/J/):=X(/J+1/) ;
00189             'END' ;
00190             'IF' A1=0.0 'AND' B1=0.0 'THEN' 'GOTO' E2 ;
00191             ZERO:=-B1/3.0/A1 ;
00192             'IF' ZERO > X(/J/) 'AND' ZERO < X(/J+1/) 'THEN'
00192             'BEGIN' T(/J/):=ABS((3.0*A1*ZERO+2.0*B1)*ZERO+C1);
00193                U(/J/):=ZERO ;
```

```
SC      SOURCE STATEMENT
00122     MEME SENS QUE ITERF;
00122     'BEGIN' 'REAL' SOMME ; 'INTEGER' I,J ;
00124        'REAL' 'ARRAY' EP(/1:ITERG/),
00124        RESULEP(/1:ITERG '/' 2/) ;
00125        SOMME:=0.0 ;
00126        'FOR' I:=1 'STEP' 1 'UNTIL' ITERG 'DO'
00126        'BEGIN' 'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
00126           SOMME:=SOMME+RF1G(/J+8*(I-1)/)*COS((J+8*(I-1))*
00126           3.1415926536/8.0) ;
00127           EP(/I/):=SOMME ;
00128        'END' ;
00129        EPSALGOR(EP,ITERG,RESULEP) ;
00130        RESULTATG:=2.0*EXP(AG*TG)/(8.0*TG)*(RESULEP(/ITERG '/' 2/)
00130           +0.5*RF1G(/0/)) ;
00131     'END' LAPLINV ;
00132     'INTEGER' I,J,K1,K2,K3 ;
00133     'REAL' A1,B1,C1,ZERO,X1,X2,L,LO ;
00134     'REAL' 'ARRAY' X,Y,M,T,U,Z(/0:NAF-1/),
00134        F,H(/1:2,0:2**PF/),FS,HS(/1:2,0:2**QF-1/),RF1F(/0:2**QF/);
00135   'PROCEDURE' COEFSPLINETROIS(N,X,Y,M);
00136   'VALUE' N,X,Y; 'INTEGER' N; 'ARRAY'X,Y,M;
00139   'BEGIN' 'INTEGER' I; 'ARRAY'D(/1: N-1/) ;  'REAL' A,B,C,E ;
00142        'FOR' I:=N-1 'STEP' -1 'UNTIL' 1 'DO'
00142      'BEGIN' A:=X(/I+1/)-X(/I/);
00143      B:=X(/I/)-X(/I-1/); C:=Y(/I+1/)-Y(/I/);
00145          E:=Y(/I/)-Y(/I-1/); 'IF' I=N-1 'THEN'
00146            'BEGIN' D(/I/):=(X(/I+1/)-X(/I-1/))/3;
00147          M(/I/):=C/A-E/B
00147          'END' 'ELSE' 'BEGIN' D(/I/):=(12*D(/I+1/)*(X(/I+1/)-X(/I-1/))
00147            -A*A)/(36*D(/I+1/));
00148              M(/I/):=C/A-E/B-A*M(/I+1/)/(6*D(/I+1/))
00148                 'END'
00149      'END' ;  M(/0/):=M(/N/):=0.0 ;
00150      'FOR' I:=1 'STEP' 1 'UNTIL' N-1 'DO'
00150      'IF' I=1 'THEN' M(/I/):=M(/I/)/D(/I/) 'ELSE'
00150          M(/I/):=(6*M(/I/)-(X(/I/)-X(/I-1/))*M(/I-1/))/(6*D(/I/))
00150        'END' COEFSPLINETROIS ;
00151     K1:=2**PF ; K3:=2**QF-1 ;
00153     'FOR' I:=1 'STEP' 1 'UNTIL' NTF 'DO'
00153     'BEGIN' 'FOR' J:=0 'STEP' 1 'UNTIL' NAF-1 'DO'
00153        'BEGIN' X(/J/):=AF(/J/)/TF(/I/) ;
00154              LO:=16.0*TF(/I/)/K1;
00155              'FOR' K2:=1 'STEP' 1 'UNTIL' K1 'DO'
00155              'BEGIN' L:=K2*LO;
00156                 F(/2,K2/):=H(/2,K2/):=0.0;
00157                 F(/1,K2/):=EXP(-X(/J/)*L)*SOURTIEF(L);
00158                 H(/1,K2/):=EXP(-X(/J/)*L)*ENTREEF(L);
00159              'END';
```

```
SC      SOURCE STATEMENT
00235          ININTEGER(0,T(/I/)) ;
00236       'COMMENT' LECTURE DES VALEURS DE T ;
00236       ININTEGER(0,P);
00237       ININTEGER(0,Q);
00238       OUTSTRING(1,'('P=')');
00239       OUTINTEGER(1,P);
00240       SYSACT(1,2,2);
00241       OUTSTRING(1,'('Q=')');
00242       OUTINTEGER(1,Q);
00243       SYSACT(1,2,2);
00244       LAPLACEDECONVOL(ENTREE,SORTIE,T,NT,A,NA,ITER,RESULTAT,P,Q);
00245       'FOR' I:=1 'STEP' 1 'UNTIL' NT 'DO'
00245       'BEGIN' 'COMMENT' TABULATION DES VALEURS DE T ET DE F(T) ;
00245          OUTSTRING(1,'('T=')') ;
00246          OUTREAL(1,T(/I/)) ;
00247          OUTSTRING(1,'('F(T)=')') ;
00248          OUTREAL(1,RESULTAT(/I/)) ;
00249          SYSACT(1,2,2);
00250       'END' ;
00251    'END' ;
00252    'END' ;
```

```
SC      SOURCE STATEMENT
00194               'END' ;
00195             G2: 'END' ;
00196             'FOR' J:=0 'STEP' 1 'UNTIL' NAF-3 'DO'
00196               'IF' T(/J+1/) > T(/J/) 'THEN'
00196               'BEGIN' U(/J+1/):=U(/J/) ;
00197                 T(/J+1/):=T(/J/) ;
00198               'END' ;
00199             E1:'FOR' K2:=0 'STEP' 1 'UNTIL' K1 'DO'
00199               'BEGIN' L:=K2*LO;
00200                 F(/2,K2/):=H(/2,K2/):=0.0;
00201                 F(/1,K2/):=EXP(-U(/NAF-2/)*L)*SORTIEF(L);
00202                 H(/1,K2/):=EXP(-U(/NAF-2/)*L)*ENTREEF(L);
00203               'END';
00204               TECTG(F,PF,FR,QF,16.0*TF(/I/),-1.0);
00205               TECTG(H,PF,HR,QF,16.0*TF(/I/),-1.0);
00206               'FOR' K2:=0 'STEP' 1 'UNTIL' K3 'DO'
00206                 RF1F(/K2/):=(FR(/1,K2/)*HR(/1,K2/)+FR(/2,K2/)*
00206                   HR(/2,K2/))/(HR(/1,K2/)*HR(/1,K2/)+HR(/2,K2/)*
00206                   HR(/2,K2/));
00207             LAPLINV(RF1F,TF(/I/),ITERF,U(/NAF-2/),Y(/NAF-2/)) ;
00208           RESULTATF(/I/):=Y(/NAF-2/) ;
00209           'END' ;
00210       'END' ;
00211   'END' LAPLACEDECONVOL;
00212       'REAL''PROCEDURE' ENTREE(X);'REAL'X;
00214       'BEGIN' ENTREE:=1.0+X;
00215       'END';
00216       'REAL''PROCEDURE' SORTIE(X);'REAL'X;
00218       'BEGIN' SORTIE:=X*X*X/6.0+X*X/2.0;
00219       'END';
00220   'INTEGER' NA,NT,P,Q ;
00221   ININTEGER(0,NA) ; 'COMMENT' LECTURE DU NOMBRE DES AI ;
00222   ININTEGER(0,NT) ; 'COMMENT' LECTURE DU NOMBRE DES VALEURS DE T ;
00223   'BEGIN' 'INTEGER' I,ITER ;
00224     'REAL' 'ARRAY' A(/0:NA-1/),T,RESULTAT(/1:NT/);
00225     ININTEGER(0,ITER) ;
00226     OUTSTRING(1,'(' ITER=')');
00227     OUTINTEGER(1,ITER);
00228     SYSACT(1,2,2);
00229         OUTSTRING(1,'('VALEUR DES AI')');
00230       SYSACT(1,2,2);
00231     'FOR' I:=0 'STEP' 1 'UNTIL' NA-1 'DO'
00231     'BEGIN' INREAL(0,A(/I/));
00232         OUTREAL(1,A(/I/));
00233       SYSACT(1,2,2);
00234     'END';
00235       'COMMENT' LECTURE DES AI ;
00235     'FOR' I:=1 'STEP' 1 'UNTIL' NT 'DO'
```